MAHA BARATHI ENGINEERING COLLEGE
CHINNASALEM – 606 201

Department of Computer Science and Engineering

LAB MANUAL

SUBJECT CODE	:	CS3362
SUBJECT NAME	:	C PROGRAMMING and DATA STRUCTURES
YEAR/ SEMESTER	:	II / III
REGULATION	:	2021

PREPARED BY,
J.SRIDEVI (AP/CSE)

APPROVED BY,
Mr. N. KATHIRKUMAR (HOD / CSE)

CS3362 CPROGRAMMINGANDDATASTRUCTURESLABORATORY L T PC

0 0 3 1.5

COURSEOBJECTIVES:

- □ Todevelopapplications inC
- □ Toimplementlinearandnon-lineardatastructures
- □ Tounderstandthedifferentoperationsofsearchtrees
- □ Togetfamiliarizedtosortingandsearchingalgorithms

LISTOFEXPERIMENTS

1. PracticeofCprogrammingusingstatements,expressions, decisionmakinganditerativestatements
2. PracticeofCprogrammingusingFunctionsandArrays
3. ImplementCprograms usingPointersandStructures
4. ImplementCprogramsusingFiles
5. DevelopmentofrealtimeCapplications
6. Arrayimplementationof ListADT
7. Arrayimplementation ofStackandQueueADTs
8. LinkedlistimplementationofList,StackandQueueADTs
9. ApplicationsofList,StackandQueueADTs
10. 10.Implementation ofBinaryTreesandoperationsofBinaryTrees
11. ImplementationofBinarySearchTrees
12. Implementationofsearchingtechniques
13. ImplementationofSortingalgorithms:InsertionSort,QuickSort,MergeSort
14. ImplementationofHashing–anytwo collisiontechniques

TOTAL:45PERIODS

COURSEOUTCOMES:

Attheendofthecourse,thestudents willbeableto:

CO1:UsedifferentconstructsofCanddevelopapplications

CO2:Writefunctionstoimplementlinearandnon-lineardatastructureoperations

CO3:Suggest andusetheappropriatelinear/non-lineardatastructureoperations fora givenproblem

CO4:Applyhashfunctions thatresult inacollisionfreescenariofordatastorageandRetrieval

CO5: ImplementSortingandsearchingalgorithmsforagivenapplication

LIST OF EXPERIMENTS

S.NO	EXPERIMENTS	
1	PracticeofC programmingusingstatements, expressions,decisionmakinganditerativestatements	
2	PracticeofCprogrammingusingFunctionsandArrays	
3	ImplementCprogramsusingPointersandStructures	
4	ImplementCprogramsusingFiles	
5	DevelopmentofrealtimeCapplications	
6	Arrayimplementationof ListADT	
7	Arrayimplementationof Stack andQueueADTs	
8	LinkedlistimplementationofList,StackandQueueADTs	
9	ApplicationsofList,StackandQueue ADTs	
10	ImplementationofBinaryTreesand operationsofBinaryTrees	
11	ImplementationofBinarySearchTrees	
12	Implementationofsearchingtechniques	
13	ImplementationofSortingalgorithms:InsertionSort,QuickSort,MergeSort	
14	ImplementationofHashing–anytwocollisiontechniques	

EX.NO:1(A)

CPROGRAMMINGUSINGSTATEMENTS,EXPRESSIONSD

ATE:

AIM

TowriteaCprogram toillustratetheconceptofStatementsandExpressions.

A. AREAANDCIRCUMFERENCE OFA

CIRCLEALGORITHM

Step1: Start

Step 2: Read theinputr

Step 3: Calculate area = pi * r *

r**Step 4:** Calculate circum = 2 * pi *

r**Step 5:**print area, circum,

Step 6: Stop

PROGRAM

```
#include
<stdio.h>int
main()
{
floatradius;
doublearea,circumference;
printf("\nEntertheradiusofthecircle:
");scanf("%f",&radius);
area = 3.14 * radius *
radius;circumference=2*3.14*
radius;printf("\nArea=%lf",are
a);
printf("\nCircumference=%lf",circumference);ret
urn0;
}
```

OUTPUT

```
Entertheradiusofthecircle:3Ar
```

ea =28.260000

Circumference =18.840000

B. ADDITIONOFTWONUMBERSC=A+BA

LGORITHM

Step1: Start

Step2: Declarevariables num1,num2andsum.

Step 3: Readvaluesfornum1, num2.

Step4: Addnum1 andnum2andassigntheresulttoavariablesum.

Step5: Displaysum

Step 6: Stop

PROGRAM

```
#include
<stdio.h>int
main()

{
int number1, number2,
sum;printf("Entertwointegers
:");
scanf("%d%d",&number1,&number2);s
um= number1 +number2;
printf("Sumoftwonumbers:");
printf("%d + %d = %d", number1, number2,
sum);return0;
}
```

OUTPUT

Entertwointegers: 45

Sumof two numbers:4 + 5=9

C. SUMANDAVERAGE OFTHREENUMBERSALGORITHM

Step1: Start

Step2: Read thethreenumberlet"a","b","c"form theuser.

Step3: Declared avariable"sum"and"Avg".

Step4:sum=a+b+c;

Step 5: Avg=sum/3;

Step6: Display"sum"and"Avg".

Step7: End

PROGRAM

```
#include
<stdio.h>int
main()
{
    int
    a,b,c;float
    sum;float
    avg;
    printf("\nEnterFirstNumber:");sc
    anf("%d",&a);
    printf("\nEnterSecondNumber:");sc
    anf("%d",&b);
    printf("\nEnterThirdNumber:");sc
    anf("%d",&c);
    sum=a+b+c;printf("\nsum:
    %.2f",sum);avg=sum/3.0;
    printf("\nAverageofThreeNumbers:%.2f",avg);ret
    urn0
}
```

OUTPUT

Enter First Number:

5EnterSecondNumber:7

Enter Third Number:

9sum: 21.00

Averageof ThreeNumbers: 7.00

RESULT

ThustheCProgramtoillustrateconceptofStatementsandExpressionshasbeenexecuted.

EX.NO:1(B)

CPROGRAMMINGUSINGDECISIONMAKINGD

ATE: **(CONTROLOR BRANCHING)**

AIM

TowriteaCProgramtoillustratetheconceptofDecisionmakingandBranchingstatements.

DECISIONMAKINGDESCRIPTION

C supports decision control statements that can alter the flow of a sequence of instructions. Thesestatementshelptojumpfrom onepartoftheprogram toanotherdependingon whetheraparticularconditionis satisfied ornot.

Thesedecisioncontrolstatementsinclude:

- ifstatement,
- if–elsestatement,
- if–else–ifstatement,and
- switch-casestatement.

A. IF-ELSE STATEMENT - NUMBER IS EVEN OR ODDALGORITHM

Step1: Inputanumber N

Step2:if(N%2==0)

Step3:print“EvenNumber”

Step 4:else

Step5: Print“Oddnumber”

Step6: Exit

PROGRAM

```
#include
<stdio.h>int
main()
{
int a;
printf("\nEnterthevalueofa:");sca
nf("%d",&a);
if(a%2==0)
```

```
printf("\n%diseven",a);
```

```
else
printf("\n%disodd",a);re
turn0;
}
```

OUTPUT

```
Enterthevalueof a:6
6  iseven
Enterthevalueofa:6161
is odd
```

B. IF–ELSE–IFSTATEMENT

C Program to check whether a number is negative, positive or zero using if–else–if StatementALGORITHM

Step1: Inputanumberfromtheuser.

Step2: Ifnumberislessthanzero,thenitisanegativeinteger.

Step3: Elseifnumberis greaterthanzero,thenitis apositiveinteger.

Step4: Else,thenumberis equalto zero.

Step5: End

PROGRAM

```
#include
<stdio.h>int
main()
{
int num;
printf("\nEnteranynumber:");sc
anf("%d",&num);if(num==0)
printf("\nThevalueisequaltozero");els
eif(num>0)
printf("\nThenumberispositive");
else
printf("\nThenumberisnegative");re
turn0;
}
```

OUTPUT:

Enter any number: -

50Thenumberisnegativ

eEnteranynumber: 9

Thevalue is equalto zero

C. SWITCH–CASESTATEMENT

Checkwhetherentered characterisavowelornotusingswitch–caseStatementALGORITHM

Step1: Start

Step2: DeclarecharactertypevariablechandReadchfrom User

Step 3: IF(ch== 'a'||ch== 'A'||ch== 'e'||ch=='E'||ch== 'i'||ch == 'I'|| ch=='o'||ch=='O'||ch == 'u' ||ch == 'U')Print"Vowel"

ELSE

Print"Consonant"

Step4: Stop

PROGRAM

```
#include
<stdio.h>intmain()
{
   charc;
   printf("EnteranAlphabet\n");s
   canf("%c",&c);
   switch(c)
      {case'a':
      case'A':
      case'e':
      case'E':
      case'i':
```

```
        case'I':
        case'o':
        case'O':
        case'u':
        case'U':printf("%cisVOWEL",c);br
            eak;
        default:printf("%cisCONSONANT",c);
    }
     return0;
}
```

OUTPUT

```
EnteranAlphabete
e is
VOWELEnteran
AlphabetZ
Zis CONSONANT
```

RESULT

ThustheCProgramto illustratetheconceptofDecisionmakingandBranchingstatements.

EX.NO:1(C)

DATE: **CPROGRAMMINGUSINGITERATIVESTATEMENTSD (LOOPINGSTATEMENTS)**

AIM

TowriteaCProgramto illustratetheIterationconcepts usingLoopingStatements.

DESCRIPTION

ITERATIVEORLOOPINGSTATEMENTS

Iterativestatementsareusedto repeat theexecution ofasequenceofstatementsuntilthespecifiedexpression becomes false. C supports three types of iterative statements also known as loopingstatements. Theyare

- while loop
- do–while loop
- forloop

A. WHILELOOP

PRINTTHE FIRST‘N’NUMBERSUSINGA

WHILELOOPALGORITHM

Step1: Start

Step 2: Assigni=1

Step3: Readanumber, num

Step 4: Whilei<=num

Print

iComputei=i+1

Step 5: Stop

PROGRAM

```
#include
<stdio.h>int
main()
{
int i = 1,n;
printf("EntertheNumber");s
canf("%d",&n);while(i<=n
```

)

```
{
printf("\n%d",i);i
= i +1;
}
return0;
}
```

OUTPUT:

```
EntertheNumber101
2
3
4
5
6
7
8
9
10
```

SUMOF‘N’NUMBERS

ALGORITHM

Step1: Start

Step2: ReadthevalueofN.

Step3: Declaredavariable i=0,sum=0

Step 4: Whilei<=N

```
Sum=sum+i
i=i+1
```

Step5: Display"sum"

Step6: End

PROGRAM

Cprogramto calculate thesumof numbersupto‘n’

```
#include
<stdio.h>int
main()
{
int n,m,i=0,sum =0;
printf("\nEnterthevalueofn:");sca
nf("%d",&n);
while(i<=n)
{
sum = sum +
i;i = i +1;
}
printf("\n The sum of numbers upto %d = %d", n,
sum);return0;
}
```

OUTPUT

Enterthevalue ofn: 5

Thesum ofnumbers upto 5=15

B. DO–WHILELOOP

PRINT NUMBERS FROM 1 TO

10.ALGORITHM

Step1: Start

Step 2: Assigni=1

Step3: Readanumber, num

Step4:DO

Print i

Computei=i+1

Whilei<=num

Step5: Stop

PROGRAM

```
#include
<stdio.h>intmain()
{
int i =
1;do
{
printf("\t%d",i);i
= i +1;
}
while(i<=10);
return0;
}
```

OUTPUT

1 2 3 4 5 6 7 8 9 1

C. CALCULATETHEAVERAGEOFFIRST *'n'* NUMBERS.

```
#include
<stdio.h>int
main()
{
int n, i = 0, sum
=0;floatavg=0.0;
printf("\nEnterthevalueofn:");sca
nf("%d",&n);
do
{
sum = sum +
i;i = i +1;
}
while(i<=n);
avg=(float)sum/n;
printf("\n The sum of first %d numbers = %d",n,
```

```
sum);printf("\nTheaverageof first
%dnumbers=%.2f",n,avg);
```

```
return0;
}
```

OUTPUT

```
Enterthevalue ofn: 20
Thesum offirst 20numbers=210
Theaverageoffirst 20numbers=10.05
```

C.FORLOOP

C Program to print the first *n* numbers using for loop.ALGORITHM

Step1: Start

Step 2: Assigni=1

Step3: Readanumber, num

Step 4: For i<=num

Print i

Computei=i+1

Step5: Stop

PROGRAM

```
#include
<stdio.h>int
main()
{
int i, n;
printf("\nEnterthevalueofn:");sca
nf("%d",&n);for(i=1;i<=n;i++)
        printf("\n%d",i);r
eturn0;
}
```

OUTPUT:

Enterthevalueofn:51
2
3
4
5

RESULT

ThustheCProgramtoillustrateIterationusingLoopingStatementswasexecuted.

EX.NO:2(A)

CPROGRAMMINGUSINGFUNCTIONSD

ATE:

AIM

TowriteaCProgramtoillustratetheconcepts ofFunctions.

FUNCTIONDECLARATION

Syntaxdeclaringafunction:

return_data_typefunction_name(data_typevariable1,data_typevariable2,..);

Here, **function_name** is a valid name for the function and **return_data_type** specifies the data type ofthevalue that will be returned to the callingfunction

FunctionDefinition

Thesyntaxofafunctiondefinition canbegiven as:

return_data_typefunction_name(data_typevariable1,data_typevariable2,..)

{

.....

statements

.....

return(variable);

}

FunctionCall

Afunctioncallstatementhasthefollowingsyntax:

variable_name=function_name(variable1,variable2,...);A

LGORITHM:

Step1: StarttheProgram

Step1: Readtheinputnumber,,num"

Step 1: Call the function to check whether number is Odd or Even**Step1:**Ifnum%2is0,PrintEvennumberElsePrintOddNumber**St ep1:**StoptheProgram

PROGRAM

Tocheck whetheranumberiseven orodd usingfunctions.

```
#include<stdio.h>int
evenodd(int num);int
main()
{
int num;
printf("\n Enter the number :
");scanf("%d", &num);
evenodd(num);return0;
}
intevenodd(inta)
{
if(a%2==0)
printf("\n %d is
EVEN",a);elseprintf("\n%disO
DD",a);
}
```

OUTPUT

Enterthenumber:787

8 isEVEN

Enterthenumber:55

is ODD

RESULT

ThustheCProgramto illustratetheconceptsofFunctions hasbeenexecuted.

EX.NO:2(B)

PASSINGPARAMETERSTOFUNCTIONS

DATE:

AIM

TowriteaCProgramtopassParameterstoFunctions usingCallbyValue andCallbyReference.

PROGRAM

A. CALLBYVALUE -Toswap thevalueof twointegers.

```
#include
<stdio.h>void
swap(int a, int b);int
main()
{
        int a =
        100;intb=2
        00;
        printf("Before swapping, Values in main()
        \n");printf("a=%d,b=%d\n",a,b);
        swap(a,b);
       printf("After swapping, Values in main()
        \n");printf("a=%d,b=%d\n",a,b);
 }
 voidswap (int a,int b)
 {
        int
        temp;te
        mp=a;a=
        b;b=tem
        p;
        printf("After swapping, Values in Swap() function
        \n");printf("a=%d,b=%d\n",a,b);
 }
```

OUTPUT

```
Beforeswapping,Valuesinmain()a
=100,b=200
```

A na=200,b=100

f

t

e

r

s

w

a

p

p

i

n

g

,

V

a

l

u

e

s

i

n

S

w

a

p

(

)

f

u

n

c

t

i

o

B. CALLBYREFERENCE-SwappingofTwoNumbers

```
#include<stdio.h>
voidswap(int *a, int*b);intmain()
{
        int a=10,b =20;
        printf("Before swapping, values in
        main()\n");printf("a=%d,b=%d\n",a,b);
        swap(&a,&b);
        printf("After swapping, values in
        main()\n");printf("a=%d,b=%d\n",a,b);
}
voidswap (int *a,int *b)
{
        inttemp; temp=*a;
       *a=*b;
       *b=temp;
       printf("Afterswapping,valuesinswap()function\n");pri
       ntf("a=%d,b=%d\n",*a,*b);
       }
```

OUTPUT

```
Beforeswapping,valuesinmain()a=
10,b=20
Afterswapping,valuesinswap()functiona=
20,b=10
Afterswapping,valuesinmain()a=
20,b=10
```

RESULT

Thus the C Program to pass Parameters to Functions using Call by Value and Reference wasexecuted.

EX.NO:2(C)

CPROGRAMMINGUSINGARRAYS

DATE:

AIM

TowriteaC Programtoaddtwo m*n matricesand storeitin athird m*n matrix.

ALGORITHM

Step1:StarttheProgram

Step2:DefinethreematricesA,B,C andreadtheirrespectiverowandcolumn numbers

Step3:ReadmatricesAandB.

Step4:First,startaloopfor gettingrowelementsofAand

Step5:BSecondly,insideitagainstartaloopforcolumnof

AandBStep6:PerformadditionbyC[i][j]=A[i][j]+B[i][j]intoC[i][j]

Step7:Attheendofloop,theresult ofaddition isstoredinMatrixCStep8:

StoptheProgram

PROGRAM-Toadd twom*n matricesandstoreitinathirdm*nmatrix.

```
#include
<stdio.h>int
main()
{
        intr,c,a[100][100], b[100][100],sum[100][100],i,j;
        printf("Enterthenumberofrows:");
        scanf("%d",&r);printf("Enter
        the number
        ofcolumns:");scanf("%d",&c)
        ;
        printf("\nEnter elements of
        1stmatrix:\n");for(i=0;i<r; ++i)
  for(j =0; j<c;++j)
 {
        printf("Enterelementa%d%d:",i+1,j+1);
        scanf("%d",&a[i][j]);
  }
        printf("Enterelementsof2ndmatrix:\n");fo
```

```
r(i =0; i <r;++i)
```

```
  for(j =0; j<c;++j)
 {
    printf("Enterelementb%d%d:",i+1,j+1);scanf
    ("%d",&b[i][j]);
  }
 for(i =0;i <r; ++i)
 for(j=0;j<c;++j)
{
    sum[i][j] =a[i][j] +b[i][j];
 }
  printf("\nSumoftwomatrices:\n");
 for (i = 0; i < r;
 ++i)for(j=0;j<c;
 ++j)
{
    printf("%d ",
    sum[i][j]);if(j ==c-1)
{
      printf("\n\n");
    }}}
```

OUTPUT

```
Enter the number of rows :
2Enter the number of columns :
2Enter elements of 1st
matrix:Enterelementa11: 1
Enter element a12:
1Enter element a21:
1Enterelementa22:1
Enter elements of 2nd
matrix:Enterelementb11:2
Enter element b12:
2Enter element b21:
2Enterelementb22:2
```

Sumoftwomatrices:

3 3

3 3

PROGRAM-MATRIXMULTIPLICATION

```
#include<stdio.h>
intmain()
{
        inta[10][10],b[10][10],mul[10][10],r,c,i,j,k;
        printf("Enterthenumberofrows=");sc
        anf("%d",&r);
        printf("Enterthenumberofcolumns=");sc
        anf("%d",&c);
        printf("Enter the first matrix elements
        =\n");for(i=0;i<r;i++)
{
        for(j=0;j<c;j++)scanf
        ("%d",&a[i][j]);
}
printf("Enterthesecondmatrixelements\n");
for(i=0;i<r;i++)
{
        for(j=0;j<c;j++)scanf
        ("%d",&b[i][j]);
}
printf("Multiplyofthetwomatrixes\n");fo
r(i=0;i<r;i++)
{
for(j=0;j<c;j++)
{
mul[i][j]=0;for(k=0;k<c;k
++)mul[i][j]+=a[i][k]*b[k
][j];
}
```

```
for(i=0;i<r;i++)
{
for(j=0;j<c;j++)printf("
%d\t",mul[i][j]);printf("
\n");
}
}
```

OUTPUT

```
Enter the number of rows =
3Enter the number of columns =
3Enterthefirst matrixelements
111
222
333
Enter the second matrix
elements1 1 1
222
333
Multiply of the two
 matrixes6 6 6
12 12 12
18 18 18
```

RESULT

ThustheC Program to multiplytwom*n matriceswas executed.

EX.NO:3

CPROGRAMSUSINGPOINTERSANDSTRUCTURESD

ATE:

AIM

TowriteaCProgramtoimplementtheconceptof PointersandStructures.

A. POINTERS

Apointerisavariablethatcontainsaddressormemory locationofanothervariable.Apointerisavariable thatrepresentslocation ofadataitem, suchas variable orarrayelement.

DeclaringPointerVariables

Syntax:

data_type*ptr_name;

ALGORITHM

Step1: StarttheProgram

Step2: Readintegervariablesnum1,num2,totalandpointervariablesa,bandc.

Step3: Call thefunctiontoaddtwonumbersusingpointerreferences

Step4: Storetheresult intotaland printtheresult

Step5: StoptheProgram

PROGRAM

Addtwointegersusingpointers andfunctions.

```
#include<stdio.h>

void sum (int *, int *, int
*);voidmain()
{

intnum1, num2,total;

printf("\nEnterthefirstnumber:");sca
nf("%d",&num1);
printf("\nEnterthesecondnumber:");sca
nf("%d",&num2);
sum(&num1, &num2,
&total);printf("\nTotal=%d",to
tal);
```

```
}

voidsum (int*a, int*b,int*c)
```

{

*c=*a+*b;

}

OUTPUT

Enter the first number:

23Enter the second number:

34Total=57

B. STRUCTURES

Structureisanuser-defineddatatypethatcanstorerelatedinformation.Structureiscollectionofdifferentdata types under asinglename.

StructureDeclarationSyntax:

A structuretype isgenerallydeclared byusingthefollowingsyntax:

structstruct–name

{

data_typevar–name;

data_typevar–name;

.....

};

STUDENTDETAILSUSINGSTRUCTURESA

LGORITHM

Step1: StarttheProgram

Step2: DeclarevariablesusingStructure

Step3: Defineonestructureto holdthedetails of astudent.

Step4: Readdetailsofall Studentslikerollno,name,feesandDOB.**Step**

5: Enter the details one by one and store them in the

structure.**Step6:**Displayall the details of the student

Step7: StoptheProgram

PROGRAM
ToReadandDisplaytheInformation aboutaStudentusingStructures.

```
#include
<stdio.h>void
main()
{structstudent
{
introll_no;char
name[80];float
fees;
charDOB[80];     };
structstudent stud1;
printf("\nEntertherollnumber:");sca
nf("%d",&stud1.roll_no);printf("\n
Enter the name :
");scanf("%s",stud1.name);
printf("\n Enter the fees :
");scanf("%f",
&stud1.fees);printf("\nEntert
heDOB:");scanf("%s",stud1.
DOB);
printf("\n********STUDENT'SDETAILS*******");p
rintf("\nROLLNo.=%d",stud1.roll_no);
printf("\nNAME=%s",stud1.name);pr
intf("\n FEES = %f",
stud1.fees);printf("\nDOB=%s",stud1
.DOB);
}
```

OUTPUT

```
Entertherollnumber:01E
nter the name:
RithuEnterthefees:
45000
EntertheDOB:25–09–1991
********STUDENT"S DETAILS
```

*******ROLLNo.=01

NAME =

RahulFEES=450

00.00

DOB=25–09–1991

EMPLOYEE DETAILS USING STRUCTURESALGORITHM

Step1: StarttheProgram&DeclarevariablesusingStructure

Step2: Readdetails ofallemployeeslikeemployeename,age,phonenumberandsalary.

Step3: Displayall thedetails ofemployeesandStoptheProgram

PROGRAM:

```
structemployee
{
char name[30];
intempId;floa
tsalary;
};
voidmain()
{
   structemployeeemp;
   printf("\nEnter Employee details
   :\n");printf("Enter the Name
   :");gets(emp.name);
   printf(" Enter the ID
   :");scanf("%d",&emp.empId
   );printf("Enter the Salary
   :");scanf("%f",&emp.salary)
   ;printf("\nEntereddetailis:");

   printf("\nName:
   %s",emp.name);printf("\nId: %d"
   ,emp.empId);printf("\nSalary:%f\n"
   ,emp.salary);
}
```

OUTPUT

EnterEmployeedetails:
EntertheName:RithaniE
ntertheID:2915Enterthe
Salary:25000Entereddet
ail is:
Name:
RithaniId:291
5
Salary:25000.000000

RESULT:

ThustheCProgramtoimplementtheconceptofPointersandStructureswasexecuted.

EX.NO:4

IMPLEMENTCPROGRAMSUSINGFILESD

ATE:

AIM

TowriteaC Programtoimplement theconceptofFileHandlingto storetheEmployee information.

ALGORITHM:

Step1: StarttheProgram

Step2: Declarethevariables

Step3: CreateandOpen thefileinwritemodeusingfptr=fopen("emp.txt","w+");

Step4: Get theEmployeeinformationsuch asid, nameand salaryofemployeeasenteredbyuserfromconsoleand storein thefile

Step5: StoptheProgram.

PROGRAM

StoringEmployeeInformation usingFiles

```
#include
<stdio.h>voidmai
n()
{
   FILE
   *fptr;intid;
   char
   name[30];float
   salary;
   fptr=fopen("emp.txt","w+");/*openforwriting*/if(fp
   tr==NULL)
   {
      printf("Filedoesnotexists\n");re
      turn;
   }
   printf("Enter the
   id\n");scanf("%d",&id);fpri
   ntf(fptr,"Id=%d\n",id);print
```

```
f("Enterthename\n");
```

```
    scanf("%s",name);
    fprintf(fptr,"Name=%s\n",name);p
    rintf("Enter the
    salary\n");scanf("%f",&salary);
    fprintf(fptr,"Salary=%.2f\n",salary);f
    close(fptr);
}
```

OUTPUT

```
Entertheid
1
Enterthename
Rithu
Enter the
salary120000
```

Nowopenfile**emp.txt**fromcurrentdirectory.Itwillhavefollowinginformation.

emp.txt

```
Id=1
Name=
RithuSalary=1200
00
```

RESULT

ThustheCProgramto implementtheconcept of FileHandlingtostorethe Employeeinformationwas executed.

EX.NO:5

DEVELOPMENTOFREALTIMECAPPLICATIONSD

ATE: **STUDENTINFORMATION SYSTEM**

AIM

Towriteacprogramto implement therealtimecapplicationsforthestudent information system.

ALGORITHM

Step1: StarttheProgram

Step2: DeclarevariablesusingStructure

Step3: Defineonestructureto holdthedetails of astudent.

Step4: Readdetails ofallStudentslikerollno,name,feesandDOB.

Step5: Readthe informationofall thestudents inaclass usingarrayof structures

Step6: Displayallthe details ofthe student

Step7: StoptheProgram

PROGRAM

```
#include
<stdio.h>structstu
dent
{
        charfirstName
        [50];introll;
        floatmarks;
 }s[3];
voidmain()
{
   int i;
   printf("Enter information of
   students:\n");for(i =1; i <=3;i++) {
      printf("Enterrollnumber:");s
      canf("%d",&s[i].roll);printf(
      "Enter the name:
      ");scanf("%s",
      s[i].firstName);printf("Enter
      marks:");scanf("%f",&s[i].
```

```
    marks);
}
```

```
    printf("\nDisplaying
    Information:");for(i =1; i <=3; i++)
    {
       printf("\nRoll number:
       %d",s[i].roll);printf("\nFirstname:");
       puts(s[i].firstName);
       printf("Marks:%.1f",s[i].marks);
     }
}
```

OUTPUT

```
Enterinformation ofstudents:
Enter roll number:
1Enterthename:Rithu
Enter
marks:90Enterrollnu
mber:2
Enterthename:SatrathiE
ntermarks: 95
Enter roll number:
3Enter the name:
VijayEnter marks:
90DisplayingInformatio
n:
Roll number:
1First name:
RithuMarks:90.0
Roll number:
2Firstname:Satrath
iMarks:95.0
Roll number:
3First name:
VijayMarks:90.0
```

RESULT

ThustheCProgramtoimplementtheStudent Information systemwasexecuted.

EX.NO:6 ARRAYIMPLEMENTATIONOFLISTADTs
DATE:

AIM

TowriteaCProgramtoImplementtheconceptofListusingArray.

ALGORITHM

Step1: StarttheProgram.

Step1: GettheElementsof ListasinputArray.

Step1: PrinttheElementsofgivenarray.

Step1: StoptheProgram

PROGRAM:

```
#include
<stdio.h>voidmai
n()
{
        intarray[5];
        printf("EntertheelementsofList:\n");for(
 inti =0; i <5; ++i)
        scanf("%d",
        &array[i]);printf("Element
        sofList:");for(int i = 0; i <
        5;
        ++i)printf("\n%d",array[i])
        ;
}
```

OUTPUT:

```
EntertheelementsofList:6
 3
 9
 8
 7
 ElementsofList:6
 3
 9
 8
 7
```

RESULT

ThustheCprogramtoimplementtheList usingArraywasexecuted.

EX.NO:7(A) **ARRAYIMPLEMENTATIONOFSTACK**
DATE:

AIM

TowriteaCProgram to ImplementtheconceptofStackusingArray.

ALGORITHM

Step1: StarttheProgram

Step2: ReadtheChoice

a. Push

b. Pop

c. Displayd.

Step3: IfChoiceisPush,

1. Topushanelementintothestack,checkwhetherthetopisgreaterthanor equaltothemaximumsizeofthe stack.
2. Ifso,then returnstackisfullandelementcannotbepushedintothestack.
3. Else,incrementthetopbyoneandpushthenewelementinthenewpositionoftop.

Step4: IfChoiceisPop,

1. Topopanelementfromthestack,checkwhether thetopofthestackisequalto-1.
2. Ifso,thenreturnstackisemptyandelementcannotbepoppedfromthestack.
3. Else,decrementthetopbyone.

Step5: IfChoiceis Display, DisplaytheElements of theStack

Step6: IfChoiceisExit,StoptheProgram.

PROGRAM:

```
#include<stdio.h>
#definesize5
int
stack[size];int
top;
intfull()
{
if(top>=size-
1)return1;
elseretur
n0;
```

}

```
voidpush(intitem)
{
top++;stack[top]
=item;
}
intempty()
{
if(top==-
1)return
1;else
return0;
}
intpop()
{
int
item;item=stack
[top];top--
;return(item);
}
voiddisplay()
{
int
i;if(empty(
))
printf("Stack Is
Empty!\n");else
{
printf("ElementsofStack:");f
or(i=top;i>=0;i--)
printf("\n%d",stack[i]);
}
}
voidmain()
{
intitem,ch,top=-1;
```

```
printf("<-----StackusingArray ----- >");
```

```
        while(1)
        {
        printf("\nEnter Your
        Choice:");printf("\n1.Push\n2.Pop\n3.Display\n4.
        exit\n");scanf("%d",&ch);
        switch(ch)
        {
        case1:
        printf("EnterTheitemtobepushed:\t");sc
        anf("%d",&item);
        if(full())
        printf("Stack is
        Full!\n");else
        push(item);
        break;
        case2:
        if(empty())printf("Empty
        stack!\n");else
        {
        item=pop();
        printf("Thepoppedelementis%d\n",item);
        }
        break;cas
        e
        3:display
        ();break;c
        ase
        4:exit(0);
        }
        }

}
```

OUTPUT

<-----StackusingArray ----- >

EnterYourChoice:1.Pus
h
2.
Pop3.Dis
play4.exi
t
1

EnterTheitemtobepushed:60Ent
erYourChoice:
1.Push2.
Pop3.Dis
play4.exi
t
1

EnterTheitemtobepushed:80Ent
erYourChoice:
1.Push2.
Pop3.Dis
play4.exi
t
1

EnterTheitemtobepushed:90Ent
erYourChoice:
1.Push2.
Pop3.Dis
play4.exi
t
3

ElementsofStack:

90

80

60

EnterYourChoice:

1. Push

2. Pop

3. Display

4. exit

2

Thepoppedelementis90E

nterYour Choice:

1. Push

2. Pop

3. Display

4. exit

3

ElementsofStack:

80

60

EnterYourChoice:

1.

Push2.Po

p3.Displa

y4.exit

4

RESULT

ThustheCProgramto implementtheconcept ofStackusingArraywasexecuted.

EX.NO:7(B) **ARRAYIMPLEMENTATIONOFQUEUEADTs**
DATE:

AIM

TowriteaCProgramtoimplementtheconceptofQueueusingArray.

ALGORITHM

Step1: Start

Step2: Defineaarray*queue*ofsize*max*=5

Step3: Initialize*front*=*rear*=–1

Step4: Displayamenu listingqueueoperations

Step5: Acceptchoice

Step6: Ifchoice=1thenIf

rear <max-

1Increment

rear

Store element at current position

ofrearElse

Print Queue

FullElseIfchoice

=2thenIffront=–

1then Print

QueueemptyElse

Display current

frontelementIncrement

frontElse

Ifchoice=3then

Displayqueueelementsstartingfromfront torear.

Step7: Stop

PROGRAM

```
#include<stdio.h>
#defineSIZE5int
front =-1;
int rear = -
1;int
q[SIZE];void
insert( );void
del(
);voiddisplay()
;
void main()
{
int
choice;do
{
printf("\n
Menu");printf("\n1.Inser
t");
printf("\n2.Delete");
printf("\n3.Display");
printf("\n4.Exit");
printf("\nEnterYourChoice:");s
canf("%d",&choice);switch(ch
oice)
{
case
1:insert(
);display(
);break;ca
se 2:del(
);display(
);break;ca
se
3:display(
```

);break;ca
se4:

```
printf("Endof
Program...!!!!");exit(0);
}
}
while(choice!=4);}
voidinsert( )
{
int num;
printf("\nEntertheelementtobeinserted:");sca
nf("%d",&num);
if(rear<SIZE-1)
{
q[++rear]=num;
if(front == -
1)front=0;
}
else
{
printf("\nQueueoverflow");
}
}
void del()
{
if(front ==-1)
{
printf("\nQueueUnderflow");r
eturn;
}
else
{
printf("\nDeletedItem:%d\n",q[front]);
}
if(front==rear)
{
front=-1;
```

```
rear =-1;
}
else
{
   front=front+1;
}
}
voiddisplay()
{
int i;
if(front ==-1)
{
printf("\nQueueisempty...");r
eturn;
}
printf("ElementsofQUEUE:");for
(i = front; i<=rear;
i++)printf("\t%d",q[i]);
}
```

OUTPUT:

Menu

1. Insert
2. Delete
3. Display
4. Exit

EnterYourChoice:1

Entertheelementtobeinserted:50

Elementsof QUEUE:50Menu

1. Insert
2. Delete
3. Display
4. Exit

EnterYourChoice:1

Entertheelementtobeinserted:60Ele

mentsofQUEUE:50 60

Menu

1. Insert
2. Delete
3. Display
4. Exit

Enter Your Choice:

2DeletedItem:50Elemen

tsofQUEUE:60Menu

1. Insert
2. Delete
3. Display
4. Exit

EnterYourChoice:2

DeletedItem:60Queu

eis empty....

Menu

1. Insert

2. Delete

3. Display

4. Exit

EnterYourChoice:2

Queue

UnderflowQueueis

empty....

Menu

1. Insert

2. Delete

3. Display

4. Exit

EnterYourChoice:4E

ndofProgram. ..... !!!

RESULT

ThustheCProgram toimplement theconcept ofQueueusingArraywasexecuted

EX.NO: 8(A)

LINKEDLISTIMPLEMENTATIONOFLISTD

ATE:

AIM

TowriteaCprogramtoimplementthe Linked List.

ALGORITHM

Step 1: Declarethe requiredvariableandfunctions

Step 2: Read theoptionsusingvariablec

Step3: Checkthewhile loopconditionuntiltheconditionsbecomesfalse. ifconditionsistrueexecuteStep5 and repeat thestep 4.If acondition is false goto step6 and execute

Step 4: Usingswitchcasestatementevaluate the cvariable

Ifcase1is trueexecutethecreatefunctionsandterminatetheswitchstatement.

Ifcase2 istrueexecute

thedisplayfunctionsandterminatetheswitchstatement.Ifcase3is trueexecute the exitfunctions andterminate program

PROGRAM:

```
#include<stdio.h>
#include<stdlib.h>
structnode
{
int info;
structnode *link;
};
structnode*first=NULL,*last=NULL;v
oidcreate();
void
display();void
main()
{
intc;
printf("1ForCreations\n");p
rintf("2 For
Display\n");printf("3ForEx
```

```
it\n");while(1)
{
printf("\nEntertheChoice:");
```

```
scanf("%d",&c);
switch(c)
{
case1:create();break;
case2:display();break;
case3:exit(0);
}
}
}
voidcreate()
{
structnode*NEW=malloc(sizeof(structnode));p
rintf("\n Enter the item to be created
");scanf("%d",&NEW->info);
NEW->link=NULL;
if(first==NULL)
first=NEW;
else
last->link=NEW;
last=NEW;
}
if(first==NULL)prin
tf("\nNonode");else
{
struct node
*temp=first;printf("\nEleme
ntofList:");while(temp!=NU
LL)
{
printf("\t%d     ->",temp-
>info);temp=temp->link;
}
printf("NULL");
}
}
```

OUTPUT:

```
 1 ForCreations
2 ForDisplay
3 ForExit

EntertheChoice:1
Entertheitemtobecreated60Ent
ertheChoice: 1
Entertheitemtobecreated70Ent
ertheChoice: 1
Entertheitemtobecreated90Ent
ertheChoice: 2
ElementofList:          60      ->      70      ->      90      ->NULL
```

RESULT

ThustheCprogramtoimplementthe Linked Listwasexecutedsuccessfully.

EX.NO:8(B) **LINKEDLISTIMPLEMENTATIONOF STACK**

DATE:

AIM

TowriteaCprogramtoimplement theStackADTsusingLinked List.

ALGORITHM

Step1: Includealltheheaderfilesanddeclarealltheuserdefinedfunctions.

Step 2: Definea'Node'structurewith twomembersdata andnext.

Step 3: DefineaNodepointer'top'andsetittoNULL.

Step 4: Implement the main method by displaying

Menu**Step5:**Usepush(value)-

InsertinganelementintotheStack**Step 6:**Usepop()-Deletingan

Element from a Stack

Step 7: Usedisplay() -Displayingstackof elements

PROGRAM

```
#include
<stdio.h>structnod
e
{
int info;
structnode*ptr;
}*top,*top1,*temp;
int
topelement();voidp
ush(intdata);voidpo
p();
void
display();void
create();intmai
n()
{
int data, ch,
e;printf("1-
Push\t");
```

```
printf("2-Pop\t");
printf("3-
Top\t");printf("4-
Dipslay\t");printf("5-
Exit");create();
```

```
   while(1)
   {
   printf("\n Enter choice :
   ");scanf("%d",&ch);
      switch(ch)
      {
      case1:
         printf("Enterdatatobepushed:");sc
         anf("%d",&data);
         push(data);
         break;
      case2:
         pop();
         break;
      case3:
         if(top ==NULL)
         printf("Noelementsinstack");el
         se
         {
         e=topelement();
         printf("\nTop element:%d",e);
         }
        break;
       case4:
        display();
       break;cas
       e5:
        exit(0);
       default:
          printf("Wrongchoice,Pleaseentercorrectchoice");brea
          k;
 }     }    }
voidcreate()
          top=NULL;v
oidpush(int data)
```

```
{
   if(top ==NULL)
   {
      top=(structnode*)malloc(1*sizeof(structnode));to
      p->ptr=NULL;
      top->info=data;
   }
   else
   {
      temp=(structnode*)malloc(1*sizeof(structnode));te
      mp->ptr=top;
      temp-
      >info=data;top=
      temp;
   }}
voiddisplay()
{
 top1 = top;
   if(top1 ==NULL)
   {
printf("Stackisempty");r
          eturn;
   }
   while(top1!=NULL)
   {
printf("%d",top1-
>info);top1= top1->ptr;
   }    }
void pop()
{
   top1 =top;
   if(top1 ==NULL)
   {
      printf("\nTryingtopopfromemptystack");ret
      urn;
```

}

```
    else
      top1=top1->ptr;
      printf("\nPoppedvalue:%d",top-
      >info);free(top);
      top = top1;
}
inttopelement()
{
   return(top->info);
}
```

OUTPUT

```
1-Push2-Pop3-Top4-Dipslay5-
ExitEnterchoice: 1
Enterdatatobepushed:50En
terchoice: 1
Enterdatatobepushed:60En
terchoice: 1
Enterdatatobepushed:70En
terchoice: 3
Topelement:70
 Enterchoice:4
 70 60 50
 Enterchoice:2
 Poppedvalue:70
 Enterchoice:4
 60 50
 Enterchoice:5
```

RESULT

ThustheCprogramtoimplementtheStackADTsusingLinked Listwasexecuted.

EX.NO:8(C)

LINKEDLISTIMPLEMENTATIONOFQUEUED

ATE:

AIM

TowriteaCprogram toimplementtheQueueADTsusingLinked List.

ALGORITHM

Step 1: Define asinglylinked list nodeforQueue

Step 2: CreateHeadnode

Step 3: Displayamenu listingQueueoperations andaccept choice

Step 4: Ifchoice=1 then

Createanewnodewithdata Makene

wnodepointtofirstnode

Makeheadnodepointtonewnode Els

e If choice=2 then

Maketemp nodepoint to first node

Make head node point to next of temp node & Release

memory Else If choice=3 then

Displaystack elements starting from head nodetill null

PROGRAM

```
#include
<stdio.h>structnod
e
{
intlabel;
structnode *next;

};
voidmain()
{
int
ch=0;int
k;
structnode *h,*temp, *head;
```

```
head=(structnode*)malloc(sizeof(structnode));he
ad->next= NULL;
while(1)
```

```
{
printf("\nQueueusingLinkedList");pri
ntf("\n 1.Insert");
printf("\n2.Delete");pri
ntf("\n3.View");
printf("\n4.Exit");
printf("\nEnteryourchoice:");sca
nf("%d",&ch);
switch(ch)
{
case1:
temp=(structnode*)(malloc(sizeof(structnode)));p
rintf("EnterDatafornew node: ");
scanf("%d", &temp-
>label);h=head;
while (h->next !=
NULL)h =h->next;
h->next =
temp;temp-
>next=NULL;break;
case2:
h=head->next;
head->next = h-
>next;printf("Node
deleted \n");free(h);
break;
case3:
printf("\n\nHEAD-
>");h=head;
while(h->next!=NULL)
{
h =h->next;
printf("%d->",h->label);
}
printf("NULL\n");
```

```
break;
case4:
exit(0);
}
}
}
```

OUTPUT

```
QueueusingLinkedList
 1. Insert
2.Delete
3.View4
.Exit
 Enteryourchoice:1
 EnterData for new node :
 50Queue using Linked
 List1.Insert
 2.Delete
 3.View
 4.Exit
 Enteryourchoice:1
 EnterData for new node :
 60Queue using Linked
 List1.Insert
 2.Delete
 3.View4
 .Exit
 Enteryourchoice:1
 EnterData for new node :
 70Queue using Linked
 List1.Insert
 2.Delete
 3.View4
 .Exit
 Enteryourchoice:3
```

HEAD->50->60->70->NULL

QueueusingLinkedList1.

Insert

2.Delete

3.View4

.Exit

Enteryourchoice:2N

odedeleted

QueueusingLinkedList1.

Insert

2.Delete

3.View4

.Exit

Enteryourchoice:3

HEAD->60->70->NULL

QueueusingLinkedList1.

Insert

2.Delete

3.View4

.Exit

Enteryourchoice:4

RESULT

ThustheCprogramtoimplementtheQueueADTsusingLinked Listwasexecuted.

EX.NO:9 (A)

APPLICATIONOFLIST(POLYNOMIALMANIPULATIONS)D

ATE:

AIM

TowriteaCprogramtoapplicationoflist(polynomial manipulations)

PROGRAM

```
#include
<stdio.h>#include
<conio.h>#include<
malloc.h>structnode
{
int
num;intc
oeff;
structnode *next;
};
structnode*start1=NULL;st
ructnode*start2=NULL;stru
ctnode*start3=NULL;struct
node*start4=NULL;structn
ode*last3=NULL;
struct node *create_poly(struct node
*);structnode*display_poly(structnode*)
;
structnode*add_poly(structnode*,structnode*,structnode*);struct
node *sub_poly(struct node *, struct node *, struct node
*);structnode *add_node(struct node *, int, int);
intmain()
{
int
option;clr
scr();do
{
printf("\n******* MAIN MENU
```

*******");printf("\n 1. Enter the first polynomial");printf("\n2.Displaythe first polynomial");

```
printf("\n 3. Enter the second
polynomial");printf("\n4.Displaythesecondpo
lynomial");printf("\n5. Add
thepolynomials");
printf("\n 6. Display the
result");printf("\n 7. Subtract the
polynomials");printf("\n 8. Display the
result");printf("\n9. EXIT");
printf("\n\n Enter your option :
");scanf("%d",&option);switch(o
ption)
{
case1:start1=create_poly(start1);br
eak;
case2:start1=display_poly(start1);br
eak;
case3:start2=create_poly(start2);br
eak;
case4:start2=display_poly(start2);br
eak;
case5:start3=add_poly(start1,start2,start3);bre
ak;
case6:start3=display_poly(start3);br
eak;
case7:start4=sub_poly(start1,start2,start4);bre
ak;
case8:start4=display_poly(start4);br
eak;
}
}while(option!=9);
getch();
return0;
}
structnode*create_poly(structnode*start)
{
```

```
structnode*new_node,*ptr;
```

```
intn,c;
printf("\n Enter the number :
");scanf("%d",&n);
printf("\tEnteritscoefficient:");sc
anf("%d",&c);
while(n !=–1)
{
if(start==NULL)
{
new_node=(structnode*)malloc(sizeof(structnode));ne
w_node->num = n;
new_node -> coeff =
c;new_node-
>next=NULL;start=
new_node;
}
else
{
ptr=start;
while(ptr-
>next!=NULL)ptr =ptr-
>next;
new_node=(structnode*)malloc(sizeof(structnode));ne
w_node->num = n;
new_node -> coeff =
c;new_node-
>next=NULL;ptr->next=
new_node;
}
printf("\n Enter the number :
");scanf("%d",&n);
if(n == –
1)break;
printf("\tEnteritscoefficient:");sc
anf("%d",&c);
```

```
}
returnstart;
}
```

```
structnode*display_poly(structnode*start)
{
structnode*ptr;p
tr=start;
while(ptr!=NULL)
{
printf("\n%dx %d\t",ptr->num,ptr-> coeff);ptr
=ptr->next;
}
returnstart;
}
structnode*add_poly(structnode *start1,structnode*start2,structnode *start3)
{
structnode*ptr1,*ptr2;i
ntsum_num, c;
ptr1=start1, ptr2 =start2;
while(ptr1!= NULL&&ptr2!=NULL)
{
if(ptr1-> coeff==ptr2->coeff)
{
sum_num= ptr1->num+ptr2 ->num;
start3=add_node(start3,sum_num,ptr1-
>coeff);ptr1 =ptr1->next;
ptr2 =ptr2-> next;
}
elseif(ptr1->coeff>ptr2->coeff)
{
start3=add_node(start3,ptr1->num,ptr1->coeff);ptr1
=ptr1->next;
}
elseif(ptr1->coeff<ptr2->coeff)
{
start3=add_node(start3,ptr2->num,ptr2->coeff);ptr2
=ptr2->next;
}
```

```
}
if(ptr1==NULL)
{
while(ptr2!=NULL)
{
start3=add_node(start3,ptr2->num,ptr2->coeff);ptr2
=ptr2->next;
}
}
if(ptr2==NULL)
{
while(ptr1!=NULL)
{
start3=add_node(start3,ptr1->num,ptr1->coeff);ptr1
=ptr1->next;
}
}
returnstart3;
}
structnode*sub_poly(struct node*start1, structnode*start2, structnode *start4)
{
structnode*ptr1,*ptr2;i
ntsub_num, c;
ptr1 = start1, ptr2 =
start2;do
{
if(ptr1-> coeff==ptr2->coeff)
{
sub_num=ptr1 ->num– ptr2 ->num;
start4=add_node(start4,sub_num,ptr1-
>coeff);ptr1 =ptr1->next;
ptr2 =ptr2-> next;
}
elseif(ptr1->coeff>ptr2->coeff)
{
```

```
start4=add_node(start4,ptr1->num,ptr1->coeff);ptr1
=ptr1->next;
}
elseif(ptr1->coeff<ptr2->coeff)
{
start4=add_node(start4,ptr2->num,ptr2->coeff);ptr2
=ptr2->next;
}
}while(ptr1!=NULL||ptr2!=NULL);if(p
tr1==NULL)
{
while(ptr2!=NULL)
{
start4=add_node(start4,ptr2->num,ptr2->coeff);ptr2
=ptr2->next;
}
}
if(ptr2==NULL)
{
while(ptr1!=NULL)
{
start4=add_node(start4,ptr1->num,ptr1->coeff);ptr1
=ptr1->next;
}
}
returnstart4;
}
structnode*add_node(structnode *start,intn, intc)
{
structnode*ptr,*new_node;if
(start== NULL)
{
new_node=(structnode*)malloc(sizeof(structnode));ne
w_node->num = n;
new_node->coeff =c;
```

```
new_node-
>next=NULL;start=
new_node;
}
else
{
ptr=start;
while(ptr-
>next!=NULL)ptr =ptr-
>next;
new_node=(structnode*)malloc(sizeof(structnode));ne
w_node->num = n;
new_node -> coeff =
c;new_node-
>next=NULL;ptr->next=
new_node;
}
returnstart;
}
```

OUTPUT

```
*******MAINMENU *******
1. Enterthe first polynomial
2. Displaythefirst polynomial
9.EXIT
Enter your option:
1Enter the number:
6Enteritscoefficient:2
Enter the number:
5Enteritscoefficient:1
Enter the number: –
1Enteryouroption:2
6 x2 5x1
Enteryouroption:9
```

RESULT

ThustheCprogramtoapplicationoflist(polynomialmanipulationswasexecuted.

EX.NO:9(B) **BAPPLICATION OFSTACK**

DATE: **(CONVERSIONOFINFIXTOPOSTFIXEXPRESSION)**

AIM

TowriteaCprogram toapplicationof stack(conversion ofinfix topostfix expression)

PROGRAM

```
#include
<stdio.h>#include
<conio.h>#include
<ctype.h>#include
<string.h>#define
MAX
100charst[MAX];
inttop=-1;
void push(char st[],
char);charpop(char st[]);
void InfixtoPostfix(char source[], char
target[]);intgetPriority(char);
voidmain()
{
 char infix[100],
postfix[100];clrscr();
 printf("\nEnteranyinfixexpression:");gets(inf
 ix);
 strcpy(postfix,
 "");InfixtoPostfix(infix,postf
 ix);
 printf("\nThecorrespondingpostfix
 expressionis:");puts(postfix);
 getch();
}
void InfixtoPostfix(charsource[],chartarget[])
{
 int i=0,
 j=0;chartem
```

```
p;
strcpy(target,"");
```

```
while(source[i]!='\0')
{
 if(source[i]=='(')
 {
  push(st,
  source[i]);i++;
 }
 elseif(source[i] ==')')
 {
  while((top!=-1)&&(st[top]!='('))
  {
   target[j] =
   pop(st);j++;
  }
  if(top==-1)
  {
   printf("\nINCORRECTEXPRESSION");
   exit(1);
  }
  temp=pop(st);//removeleftparenthesisi+
  +;
 }
 elseif(isdigit(source[i])||isalpha(source[i]))
 {
  target[j] =
  source[i];j++;
  i++;
 }
 else if (source[i] == '+' || source[i] == '–' || source[i] == '*'
 ||source[i] =='/'||source[i]=='%')
 {
  while((top!=-1)&&(st[top]!='(')&&(getPriority(st[top])>getPriority(source[i])))
  {
   target[j] =
   pop(st);j++;
```

```
   }
   push(st,
   source[i]);i++;
  }
  else
  {
   printf("\nINCORRECTELEMENTINEXPRESSION");
   exit(1);
  }
 }
 while((top!=-1)&&(st[top]!='('))
 {
  target[j] =
  pop(st);j++;
 }
 target[j]='\0';
}
intgetPriority(charop)
{
 if(op=='/'||op==
  '*'||op=='%')return1;
 else if(op=='+' || op=='–
  ')return0;
}
voidpush(charst[],char val)
{
 if(top==MAX-1)
  printf("\nSTACKOVERFLOW");e
 lse
 {
  top++;st[top
  ]=val;}
}
charpop(charst[])
{
```

```
charval='';i
f(top==-1)
 printf("\nSTACKUNDERFLOW");
else
{
 val=st[top];
top--;
 }returnval;
}
```

OUTPUT

Enteranyinfixexpression:A+B*C

Thecorrespondingpostfix expression is:ABC*+

RESULT

Thus the C program to application of stack (conversion of infix to postfix expression)Was executed.

EX.NO:9(C) APPLICATION OF QUEUE (JOSEPHUS PROBLEM)DATE:

AIM

TowriteaC program toapplication ofqueue(Josephus problem).

PROGRAM

```
#include
<stdio.h>#include
<conio.h>#include<
malloc.h>structnode
{
int
player_id;structn
ode*next;
};
structnode*start,*ptr,*new_node;in
tmain()
{
int n, k, i,
count;clrscr();
printf("\nEnterthenumberofplayers:");sca
nf("%d",&n);
printf("\nEnterthevalueofk(everykth player getseliminated):
");scanf("%d",&k);
//Createcircularlinkedlistcontainingalltheplayersstart
= malloc(sizeof(struct node));
start–
>player_id=1;ptr=st
art;
for(i =2; i <=n; i++)
{
new_node=malloc(sizeof(structnode));p
tr–>next = new_node;new_node–
>player_id = i;new_node–>next=start;
ptr=new_node;
```

```
}
for(count=n; count>1;count—)
{
for(i=0;i<k–
1;++i)ptr=ptr–>next;
ptr–>next = ptr–>next–>next; // Remove the eliminated player from
thecircularlinked list
}
printf("\n The Winner is Player %d", ptr–
>player_id);getche();
return0;
}
```

OUTPUT

```
Enter thenumberof players: 5
Enterthevalueofk (everykthplayer getseliminated):
2TheWinneris Player 3
```

RESULT

ThustheC programtoapplication ofqueue(Josephusproblem)wasexecuted.

EXNO:10

IMPLEMENTATIONBINARYTREEANDOPERATIONSOFBINARYTREESDA

TE:

AIM

TowriteaCprogramImplementationBinaryTree and operationsOf BinaryTrees.

ALGORITHM

Step1: Startfromroot.

Step2: Comparethe insertingelement withroot, ifless than root,then recurseforleft,else recurseforright.

Step3: Ifelementtosearchisfoundanywhere,returntrue,else returnfalse.

PROGRAM

```
#include<stdio.h>
#include<stdlib.h>
structtree{
intdata;
struct tree
*left;structtree*r
ight;
} *root = NULL, *node = NULL, *temp =
NULL;structtree* insert(intkey,struct tree*leaf){
if(leaf == 0)
{structtree*temp;
temp=(structtree*)malloc(sizeof(structtree));te
mp->data =key;
temp->left=0;
temp->right =
0;printf("Data
inserted!\n");returntemp;
}
else{
if(key<leaf->data)
leaf->left = insert(key,leaf-
>left);else
leaf->right=insert(key,leaf->right);
```

```
}
returnleaf;
}
structtree*search(intkey,structtree*leaf){if(l
eaf !=NULL){
if(key == leaf->data)
{printf("Datafound!\n")
;returnleaf;
}
else{
if(key<leaf->data)
returnsearch(key,leaf-
>left);else
struct tree* minvalue(struct tree *node)
{if(node==NULL)
return
NULL;if(nod
e->left)
returnminvalue(node-
>left);else
returnnode;
}
/*FunctionforfindmaximumvaluefromtheTree*/struct
tree* maxvalue(struct tree*node){
if(node==NULL)r
eturn
NULL;if(node-
>right)
returnmaxvalue(node-
>right);else
returnnode;
}
voidpreorder(structtree*leaf){if(
leaf ==NULL)
return;
```

```
printf("%d\n",leaf-
>data);preorder(leaf-
>left);
```

```
preorder(leaf->right);
}
voidinorder(structtree*leaf){if(
leaf ==NULL)
return;preorder(leaf
->left);
printf("%d\n",leaf-
>data);preorder(leaf-
>right);
}
voidpostorder(structtree*leaf){if(
leaf ==NULL)
return;preorder(leaf
->left);
preorder(leaf-
>right);printf("%d\n",leaf
->data);
}
structtree*delete(struct
tree*leaf,intkey){if(leaf ==NULL)
printf("ElementNotFound!\n");e
lseif(key<leaf->data)
leaf->left=delete(leaf-
>left,key);elseif(key>leaf->data)
leaf->right=delete(leaf-
>right,key);else{
if(leaf->right && leaf->left)
{temp = minvalue(leaf-
>right);leaf->data =temp-
>data
else{
temp=leaf;
if(leaf-
>left==NULL)leaf=lea
f->right;
```

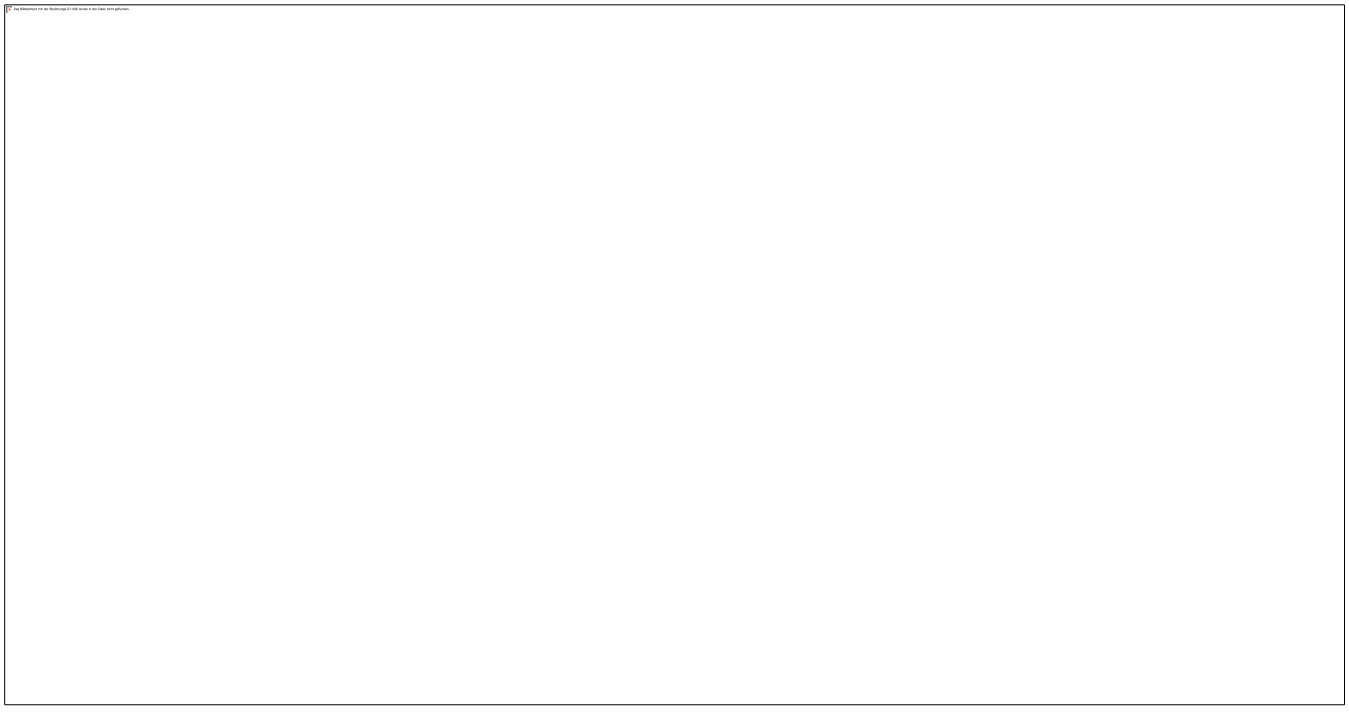
```
elseif(leaf-
>right==NULL)leaf=leaf-
>left;
free(temp);
printf("Datadeletesuccessfully!\n");
```

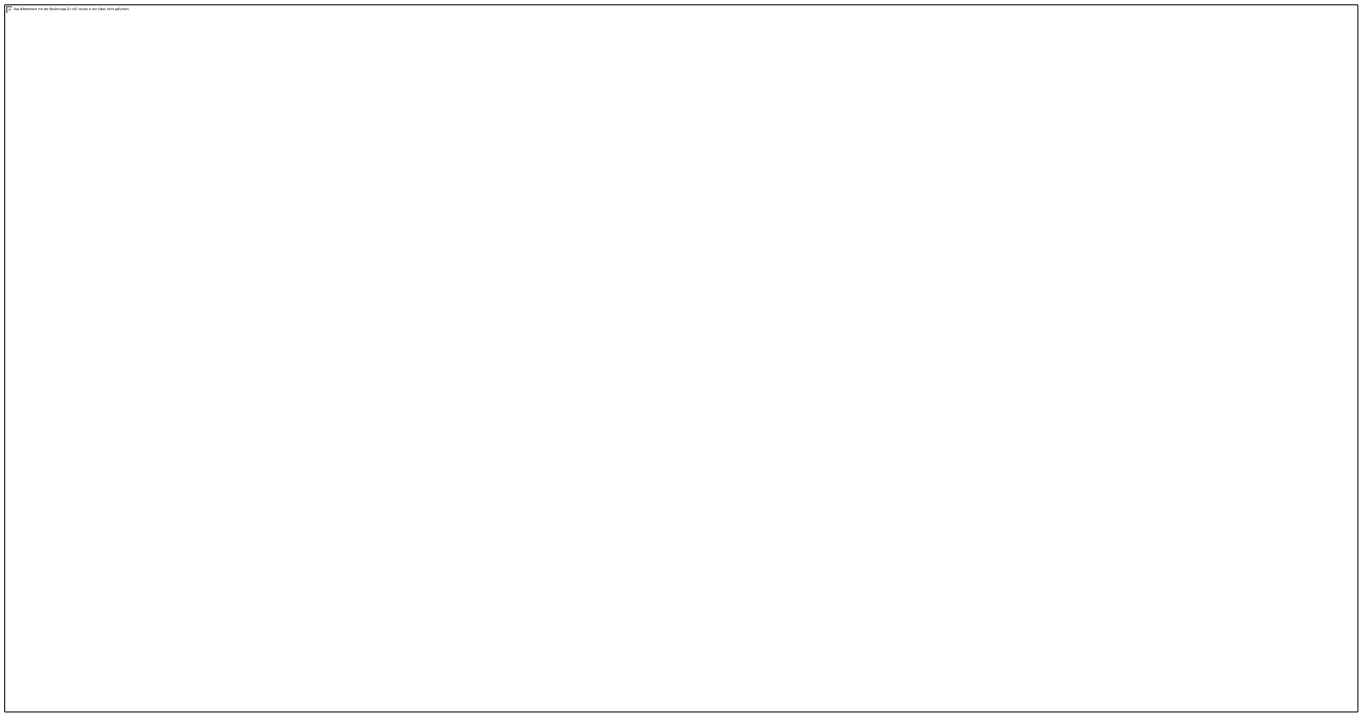
```
}
}
}
intmain() {
int key,
choice;while(choice
!=7){
printf("1. Insert\n2. Search\n3. Delete\n4. Display\n5. Min Value\n6. Max Value\n7.
Exit\n");printf("Enteryourchoice:\n");
scanf("%d",
&choice);switch(choic
e){
case1:
printf("\nEnterthevaluetoinsert:\n");sc
anf("%d",&key);
root=insert(key,root);b
reak;
case2:
printf("\nEnterthevaluetosearch:\n");sc
anf("%d",&key);
search(key,root);
break;
case3:
printf("\nEnterthevaluetodelete:\n");sc
anf("%d",&key);
delete(root,key);
break;
case
4:printf("Preorder:\n"
);preorder(root);printf
("Inorder:\n");inorder
(root);printf("Postord
er:\n");postorder(root
);break;
case5:
```

```
if(minvalue(root)==NULL)
```

```
printf("Tree is
empty!\n");else
printf("Minimumvalueis%d\n",minvalue(root)-
>data);break;
case6:
if(maxvalue(root)==NULL)p
rintf("Tree is empty!\n");else
printf("Maximumvalueis%d\n",maxvalue(root)-
>data);break;
case7:
printf("ByeBye!\n");
exit(0);
break;d
efault:
printf("Invalidchoice!\n");
}
}
return0;
```

OUTPUT





RESULT

Thustheprogram inCis implemented BinaryTreeandOperationsofBinaryTrees.

EX.NO: 11 IMPLEMENTATION OF BINARY SEARCH

TREEDATE:

AIM

Tolocate anelement in asortedarrayusingBinarysearch treemethod

ALGORITHM

Step1: Start

Step2: Read „n‟‟number of arrayin elements

Step3: Create an array*arr*consisting*n*sorted elements

Step4: Get keyelementtobesearched

Step5: Assign0 to*left* and*n-1* to*right*

While(*left* <*right*)

Find middle element *mid* =

(right+left)/2If key=arr[mid]then

Print

midStop

Elseifkey>arr[mid]thenlef

t= mid + 1

else

right=mid–1

Step6: Print"Element notfound"

Step7: Stop

PROGRAM

```
#include
<stdio.h>intmain()
 {
 inta[50],i,n,right,left,mid,key,found;printf("
 Enterarraysize:");
 scanf("%d",&n);
 printf("Enterarrayelements:");fo
 r(i=0;i<n; i++)
```

```
scanf("%d",&a[i]);
printf("\n Elements in Sorted Order
\n");for(i=0;i<n; i++)
printf("%d\t",a[i]);
printf("\nEnterelementtolocate:");sca
nf("%d",&key);
right=n-
1;left= 0;
found=-1;
while(left<=right)
{
mid=(right+left)/2;if(
a[mid] ==key)
{
printf("Locatedatposition%d",mid);fo
und=1;
break;
}
elseif(a[mid]>key)ri
ght= mid -1;
else
left= mid+ 1;
}
if (found == -
1)printf("Elementnotfound"
);return0;
```

OUTPUT

Enter array size:
5Enterarrayelements:
10
20
40
50
70
ElementsinSortedOrder
10 20 40 50 70
Enter element to locate:
70Locatedat position4

OUTPUT2:

Enter array size:
6Enterarrayelements:
10
20
30
40
50
60
ElementsinSortedOrder
10 0 0 40 50 60
Enterelementtolocate:70El
ementnot found

RESULT

Thus an element is located quicklyusingbinarysearch method.

EX.NO:12(A)

IMPLEMENTATIONOFLINEARSEARCH

DATE:

AIM

Toperformlinearsearchofanelementonthegivenarray.

ALGORITHM

Step1: Start
Step2: Readnumber ofarrayelements*n*
Step3: Readarrayelements A_i,$i = 0,1,2,\ldots n-1$
Step4: Read*search*key
Step5: Assign0to*found*
Step6: Checkeacharrayelementagainst*search*
IfA_i=*search*then
found = 1
Print "Element
found"Print position*i*
Stop
Step7: If*found*= 0then
print"Elementnotfound"
Step8: Stop

PROGRAM

```
#include<stdio.h>
intmain()
{
inta[50],i,n,key,found;
printf("Enternumberofelements:");sc
anf("%d",&n);
printf("EnterArrayElements:\n");fo
r(i=0;i<n;i++)scanf("%d",&a[i]);
printf("Enterelementtobesearched:");sca
nf("%d",&key);
```

```
found =
0;for(i=0;i<n;i+
+)
{
if(a[i]==key)
{
printf("Elementfoundatposition%d",i);fo
und=1;
break;
}
}
if(found==0)
printf("\nElementnotfound");r
eturn0;
}
```

OUTPUT1:

Enter number of elements:
5EnterArrayElements:
30
20
10
50
60
Enterelementtobesearched:50Ele
mentfoundat position3

OUTPUT2:

Enter number of elements:
5EnterArrayElements:
1
2
4
5
8
Enter element to be searched:
3Elementnot found

RESULT:

Thusthelinearsearchofanelementonthegivenarraywasperformedandoutputwasverified.

EX.NO:12(B)

IMPLEMENTATIONOFBINARYSEARCHD

ATE:

AIM

Tolocateanelement ina sorted arrayusingBinarysearchmethod

ALGORITHM

Step1:Start

Step 2: Read,,n‟‟numberofarrayinelements

Step3: Createanarray*arr*consisting*n* sorted elements

Step 4: Get key element to be

searched**Step 5:** Assign 0 to *left* and *n-*

*1*to *right***Step6:** While(*left*<*right*)

Find middle element *mid* =

(right+left)/2If key=arr[mid]then

Print

midStop

Elseifkey>arr[mid]thenlef

t=mid +1

else

right=mid –1

Step7: Print "Elementnotfound"

Step8: Stop

PROGRAM

```
#include<stdio.h>in
tmain()
{
inta[50],i,n,right,left,mid,key,found;print
f("Enterarraysize:");
scanf("%d",&n);
printf("Enterarrayelements:");fo
r(i=0; i<n;
i++)scanf("%d",&a[i]);
printf("\nElementsinSortedOrder\n");
```

```
for(i=0; i<n;
i++)printf("%d\t",a[i]);
printf("\nEnterelementtolocate:");sca
nf("%d",&key);
right=n-
1;left=0;
found= -1;
while(left<=right)
{
mid=(right+left)/2;if
(a[mid]==key)
{
printf("Locatedatposition%d",mid);fo
und= 1;
break;
}
elseif(a[mid]>key)ri
ght=mid-1;
else
left=mid+1;
}
if(found== -
1)printf("Elementnotfound"
);return0;
}
```

OUTPUT1:

Enter array size:

5Enterarrayelements:

10

20

40

50

70

ElementsinSortedOrder

10 20 40 50 70

Enterelementtolocate:70L

ocatedat position 4

OUTPUT2:

Enter array size:

6Enterarrayelements:

10

20

30

40

50

60

ElementsinSortedOrder

10 0 0 40 50 60

Enterelementtolocate:70El

ementnot found

RESULT

Thus an element islocated quicklyusingbinarysearch method.

EX.NO:13(A)

IMPLEMENTATIONOFINSERTIONSORT

DATE:

AIM

Tosort an arrayofNnumbersusing Insertion sort.

ALGORITHM

Step1: Start

Step2: Read numberof arrayelements*n*

Step3: Read arrayelements *Ai*

Step4: Sorttheelementsusinginsertion sort

- Inpassp,movetheelementinpositionpleftuntilitscorrectplaceis foundamongthe first p + 1 elements.
- Elementatpositionpissavedintemp,andall largerelements (priortopositionp)aremovedonespot to theright. Thentemp is placedinthecorrectspot.

Step5: PrinttheSortedarrayusingFor Loop.

Step6: Stop

PROGRAM:

```
#include
<stdio.h>intmain()

{
int i, j, k, n, temp, a[20],

p=0;printf("Entertotalelements

:");scanf("%d",&n);

printf("Enterarrayelements:");for(i=

0;i<n; i++)

scanf("%d",

&a[i]);for(i=1;i<n;i

++)

{
temp =

a[i];j = i -1;
```

```
 while((temp<a[j])&&(j>=0))
{
 a[j+1]=a[j];
```

```
 j = j -1;

}
 a[j+1] =

 temp;p++;

 printf("\nAfterPass%d:",p);for

 (k=0;k<n; k++)

 printf("%d",a[k]);
 }
 printf("\nSortedList:");fo

 r(i=0;i<n;i++)printf("%d

 ",a[i]);

 return0;

 }
```

OUTPUT:

```
Entertotalelements: 6
Enter arrayelements:3 6 1 2 8 5
AfterPass1:    3 6 1 2 8 5
AfterPass2:    1 3 6 2 8 5
AfterPass3:    1 2 3 6 8 5
AfterPass4:    1 2 3 6 8 5
AfterPass5:    1 2 3 5 6 8
SortedList:    1 2 3 5 6 8
```

RESULT

Thusarrayelements aresortedusingInsertionSort.

EX.NO:13(B)

IMPLEMENTATIONOFQUICKSORTD

ATE:

AIM

Tosort an arrayof Nnumbers usingQuick sort.

ALGORITHM

Step1: StarttheProgram.

Step2: Readnumberofarrayelements*n and*arrayelements*Ai***Step**

3: Pick an element from an array, call it as pivot

element.**Step4:**Divideanunsorted arrayelementintotwoarrays.

Step5: Ifthevaluelessthanpivot

elementcomeunderfirstsubarray,theremainingelementswithvalue greaterthan pivot come in secondsub array.

Step6: Print theSortedarrayandstop.

PROGRAM

```
#include<stdio.h>
voidquicksort(int num[25],intfirst,intlast)
{
  int i, j, pivot,
  temp;if(first<last){
    pivot=first;
    i=first;j=la
    st;while(i<
    j)
{
      while(num[i]<=num[pivot]&&i<last)i
      ++;
      while(num[j]>num[pivot])
      j--;
      if(i<j)
      {
        temp=num[i];n
        um[i]=num[j];n
        um[j]=temp;
      }
```

}

```
    temp=num[pivot];num[
    pivot]=num[j];num[j]=t
    emp;quicksort(num,firs
    t,j-
    1);quicksort(num,j+1,la
    st);
  }
}
int main()
{
  int i, n, num[25];
  printf("Enterthetotalnumberofelements:");sca
  nf("%d",&n);
  printf("Enter%delements:",n);fo
  r(i=0;i<n;i++)scanf("%d",&nu
  m[i]);quicksort(num,0,n-1);
  printf("Sortedelements:");f
  or(i=0;i<n;i++)
  printf("
  %d",num[i]);return0;
}
```

OUTPUT

```
Enterthetotalnumberofelements:6Ent
er6 elements:
3
5
9
1
0
8
Sortedelements:0 13 58 9
```

RESULT

Thusarrayelementsaresorted usingQuick sort.

EX.NO:13(C)

IMPLEMENTATIONOFMERGESORTD

ATE:

AIM

To sort an arrayof Nnumbers usingmergesort.

ALGORITHM

Step1:Start

Step2:Declarearrayand

left,right,midvariableStep3: Perform

mergefunction.

mergesort(array,left,right)

mergesort(array,left,right)if

left > right

return

mid=

(left+right)/2mergesort(array,

left,

mid)mergesort(array,mid+1,ri

ght)merge(array,left,mid,right

)

Step4: Stop

PROGRAM

```
#include<stdio.h>
voidmerge(inta[],int b[],intc[],intp,intq)
{
    int
    i=0,j=0,k=0;while((i
    <p)&&(j<q))
    {
        if(b[i]<=c[j])
            a[k++]=b[i++];
        else
    }
            a[k++]=c[j++];
```

```
if(i==p)
```

```
       while(j<q)a[k++]
           =c[j++];
      else
        while(i<p)a[k++]
           =b[i++];
       printf("\n");
 }
voidmergesort(inta[],intn)
{
int
i,j,s,b[20],c[20];s=
n/2;
if(n>1)
{
for(i=0;i<s;i++)b[i]=a[i
];for(i=s,j=0;i<n;i++,j+
+)c[j]=a[i];mergesort(b
,s);mergesort(c,n-
s);merge(a,b,c,s,n-s);
} }
int main()
{
int a[50],i,n;
printf("\tMerge Sort Using Divide & Conquer
Method\n");printf("
");printf("\nEnte
r The Number Of Elements : ");scanf("%d",&n);
```

```
printf("\nEnterTheElementsOneByOne:
");for(i=0;i<n;i++)
scanf("%d",&a[i]);
mergesort(a,n);
printf("\nTheSortedElementsAre");for
(i=0;i<n;i++)
printf("\n%d",a[i]);
return1;
}
```

OUTPUT:

```
MergeSort UsingDivide&Conquer Method
---------------------------------------------
EnterTheNumberOfElements:5Ent
erTheElements OneByOne:
6
7
1
2
9
TheSortedElementsAre1
2
6
7
9
```

RESULT

ThustheSortingofelements usingMergesort hasbeen executedsuccessfully.

Ex. No:14 **HASHING**

TECHNIQUESDATE:

AIM

TowriteaCprogram toimplement hashtable

DESCRIPTION

Hashingis atechnique toconvert a rangeof keyvalues into arangeofindexes of anarray. We're goingto use modulo operator to get a range of key values. Consider an example of hash table of size 20, andthefollowingitems areto bestored.Item areinthe(key,value) format.

ALGORITHM

Step1: Create astructure,data(hashtable item) withkeyandvalue asdata.

Step 2: Now create an array of structure, data of some certain size (10, in this case). But, the size ofarray must be immediately updated to a prime number just greater than initial array capacity(i.e10, in this case).

Step3: Amenu isdisplayedonthescreen.

Step4: User mustchooseoneoptionfrom four choices given inthemenu

Step5: Performalltheoperations

Step6: Stoptheprogram

PROGRAM

```
#include<stdio.h>
#include<stdlib.h>
structdata
{
int
key;intva
lue;
};
structdata*array;i
nt    capacity    =
10;int size=0;
/*this functiongives auniquehash codetothegiven
key*/inthashcode(int key)
{
return(key%capacity);
}
/* it returns primenumber just greaterthan
arraycapacity*/intget_prime(int n)
{
if (n %2 ==0)
{
```

```
n++;
}
for (; !if_prime(n); n +=
2);returnn;
}
/*tocheckifgiveninput
(i.en)isprimeornot*/intif_prime(int n)
{
int i;
if(n ==1||n==0)
{
return0;
}
for(i =2; i<n; i++)
{
if (n %i ==0)
{
return0;}}
return1;
}
voidinit_array()
{
int i;
capacity=get_prime(capacity);
array=(struct data*)malloc(capacity*
sizeof(structdata));for(i =0; i <capacity; i++)
{
array[i].key=0;
array[i].value=0;
}}
/* to insert a key in the hash table
*/voidinsert(int key)
{
intindex =hashcode(key);
```

```
if(array[index].value==0)
{
/* key not present, insert it
*/array[index].key =
key;array[index].value =
1;size++;
printf("\nKey(%d)hasbeeninserted\n",key);
}
elseif(array[index].key==key)
{
/* updatingalreadyexistingkey*/
printf("\nKey(%d)alreadypresent,henceupdatingitsvalue\n",key);array[
index].value+=1;
}
else
{
/*keycannotbe insertastheindexisalreadycontainingsomeother
key*/printf("\nELEMENTCANNOT BEINSERTED\n");
}}
/* to remove a key from hash table
*/voidremove_element(intkey)
{
int index =
hashcode(key);if(array[ind
ex].value==0)
{
printf("\nThis keydoesnotexist\n");
}
else
{array[index].key=0;
array[index].value =
0;size--;
printf("\nKey(%d)hasbeenremoved\n",key);}}
/*todisplayalltheelementsofahashtable*/voidd
isplay()
```

```
printf("\narray[%d]haselements -:\nkey(%d)andvalue(%d) \t",i,array[i].key,array[i].value);
}}}
intsize_of_hashtable()
{
returnsize;
}
voidmain()
{
int choice, key, value, n,
c;init_array();
do {
printf("\n Implementation of Hash Table in C
\n\n");printf("MENU-:
\n1.InsertingitemintheHashTable""\n2.Removingitem
fromtheHashTable""\n3.Checkthe sizeof Hash Table"
"\n4.DisplayaHash Table"
"\n\nPleaseenteryourchoice-
:");scanf("%d",&choice);switch(c
hoice)
{
case1:
printf("InsertingelementinHashTable\n");pr
intf("Enterkey-:\t");
scanf("%d",&key);
```

```
insert(key);
break;
case2:
printf("DeletinginHashTable\nEnterthekeytodelete-
:");scanf("%d",&key);
remove_element(key);
break;
case3:
n=size_of_hashtable();
printf("Size of Hash Table is-:%d\n",
n);break;
case4:
display();
break;def
ault:
printf("WrongInput\n");}
printf("\n Do you want to continue-:(press 1 for
yes)\t");scanf("%d",&c);
}while(c==1);
getch();
```

OUTPUT

RESULT:

ThustheCprogramtoimplemented Hash Table.

CONTENTBEYONDSYLLABUS

1. SIGLELINKEDLISTOPERATIONS

AIM

ToWriteaCprogramtoperformvariousoperationssuchascreation,insertion,deletion,search and displayon single linked list .

ALGORTHIM

Step1: Start

Step2: Declarearrayandleft,right, midvariable Step

Step 3: Performsinglylinked list.

creation(array,left,right)d

eletion (array, left,

right)ifleft >right

return

mid=

(left+right)/2mergesort(array,

left, mid)isertiont(array,

mid+1,

right)creation(array,left,mid,ri

ght)

Step4: Stop

PROGRAM

```
#include<stdio.h>
#include<conio.h>
#include<alloc.h>
#include<stdlib.h>
voidcreate();
void
insert();void
del();void
display();struc
tnode
{
intdata;
```

```
structnode *link;
};
structnode*first =null,*last =null,*next, *curr,*prev;
```

```
intch;
voidmain()
{
clrscr();
printf(“singlylinkedlist\n”);d
o
{
printf(“\n 1.create \n 2.insert \n 3.delete \n 4.exit \n
”);printf(“Enteryour choice”);
scanf(“%d”,&ch);
switch(ch)
{
case1:create();d
isplay();break;
case2:insert();d
isplay();break;
case3:del();d
isplay();brea
k;
case4:exit(0);
}
}
while(ch<=3);
}
voidcreate(){
curr=(structnode*)malloc(sizeof(structnode));pri
ntf(“Enterthedata: ”);
scanf(“%d”, &curr ->
data);curr->link =null;
first =
curr;last=c
urr;
}
voidinsert()
```

```
{
int pos, c=1;
curr=(structnode*)malloc(sizeof(structnode));p
rintf("Enterthedata:");
scanf("%d", &curr ->
data);printf("Enter the
position:");scanf("%d",&po
s);
if((pos==1)&&(first !=null))
{
curr-
>link=first;first=
curr;
}
else
{
next =
first;while(c<
pos)
{
prev = =
first;while(c<pos)
{
prev=next;
next=prev-
>link;c++;
}
if(prev==null)
{
printf("\nInvalidposition");
}
else
{
curr -> link = prev ->
link;prev->link=curr;
```

```
if(curr->link==null)
{
last=curr;
```

```
}
}
}
voiddel()
{
int pos, c=1;
printf(“Entertheposition”);s
canf(“%d”,&pos);
if(first=null)
{
printf(“\nlistisempty”);
}
elseif(pos==1)&&(first->link==null)
{
printf(“\n Deleted element is %d \n”, curr ->
data);free(curr);
}
else
{
next =
first;while(c<
pos)
{
prev=next;
next = next ->
link;c++;
}
prev -> link = next ->
link;next ->link =null;
if(next =null)
{
printf(“\nInvalidposition”);
}
else
{
```

```
printf(“\nDeletedelementis:%d\n”,next ->data);
```

```
printf(“\n Deleted element is:%d\n”, next ->
data);free(next);
if(prev->link ==null)
{
last=prev;
}
}
}
}
voiddisplay()
{
curr =
first;while(curr!=n
ull)
{
printf(“\n%d”,curr ->data
}
}
```

OUTPUT

Singlylinkedlist1
.create
2.inser
t
3.del4.
exit
Enteryourchoice1E
nter the
data:21.create
2.insert
3.del4.
exit
Enter your choice
2Enter the data:
4Enter the position:
22
4
1.create
2.insert
3.del4.
exit
Enterachoice4

RESULT

ThusthesinglyLinked Listwasexecutedsuccessfully.

2. IMPLEMENTATIONOFPRIMSALGORITHM

AIM

Towriteacprogramto implementprims algorithm

ALGORITHM

Step 1: Startfrom anyarbitraryvertex.

Step2: Notedown alltheedgesemergingfromthis vertex.

Step3: Markthis edge asvisited.

Step4: Selectanedgewiththeminimumweight.

Traversetotheotherend. Removethis edgefromthelist andinsert itinto theminimumspanningtree.

Step5: Repeat thisprocess forthenewlyvisited vertex.

Step 6: Each time you visit a vertex, check if it was already visited, only then we do the processofaddingits edges to thelist and pickingthe minimum.

Step7: Ifnot, thensimplypick upthenextminimum edge.

Step8: Repeatthisprocessuntil allthenodesarevisited.

PROGRAM

```
#include<stdio.h>
#include<conio.h>
int                    n,
cost[10][10];void
prim(){
  inti,j,startVertex,endVertex;
  intk,nr[10],temp, minimumCost=0,tree[10][3];
  /*Forfirstsmallestedge*/te
  mp =cost[0][0];
  for (i = 0; i < n; i++)
    {for(j=0;j<n;j++){
      if(temp >cost[i][j]) {
        temp =
        cost[i][j];startVe
        rtex =
        i;endVertex=j;
      }
    }
```

```
}
/*Nowwehavefistsmallestedgein
graph*/tree[0][0] =startVertex;
tree[0][1] =
endVertex;tree[0][2] =
temp;minimumCost =
temp;for(i =0; i <n;
i++) {
  if(cost[i][startVertex]<cost[i][endVertex])n
    r[i] =startVertex;
  else
    nr[i]=endVertex;
}
nr[startVertex]=100;
nr[endVertex] =100;
temp =99;
for (i = 1; i < n - 1; i++)
  {for(j =0; j<n; j++){
    if(nr[j]!=100&&cost[j][nr[j]]<temp){temp
      =cost[j][nr[j]];
      k =j;
    }
  }
   [i][0] =k;
  tree[i][1] =
  nr[k];tree[i][2]=cost[k][nr
  [k]];
  minimumCost=minimumCost+cost[k][nr[k]];nr
  [k]=100;
  for(j =0; j<n; j++){
    if(nr[j] !=100&&cost[j][nr[j]]
      >cost[j][k])nr[j]=k;
  }
  temp =99;
}
```

```
printf("\nTheminspanningtreeis:-
");for(i =0; i <n -1; i++)
```

```
{
    for (j = 0; j < 3;
      j++)printf("%d",tree[i]
      [j]);
    printf("\n");
  }
  printf("\nMincost:%d",minimumCost);
}
voidmain()
 {
  int i,
  j;clrscr(
  );
  printf("\nEntertheno.ofvertices:");sc
  anf("%d",&n);
  printf("\nEnterthecostsofedgesinmatrixform:");for(i
  =0; i <n; i++)
    for (j = 0; j < n; j++)
      {scanf("%d",&cost[i][j])
      ;
    }
  printf("\nThematrixis:");fo
  r(i =0; i <n;i++) {
    for (j = 0; j < n; j++)
      {printf("%d\t",cost[i][j])
      ;
    }
    printf("\n");
  }
  prim();
  getch();
}
```

OUTPUT

Entertheno. ofvertices :3

Enter the costs of edges in matrix form:-

99 2 3

2 99 5

3 5 99

The matrix is:-

99 2 3

2 99 5

3 5 99

The min spanning tree is:-

0 1 2

2 0 3

Mincost:5

RESULT

Thusthecprogramforprims algorithmis executedsuccessfully.