PROGRAM NO: 1

AIM: Simple program using formatted I/O.

ALGORITHM

Step 1 : start

Step 2 : read values of a and b

Step 3 : print the values of a and b

Step 4 : read value of n

Step 5 : print the value of n, right justified in the field

of 6 columns

Step 6 : print the value of n, right justified in the field

of 2 columns

Step 7 : print the value of n, left justified in the field

of 6 columns

Step 8 : print the value of n, placing zeros before n in the field

of 6 columns

Step 9 : read values of q and r

Step 10 : read value of w

Step 11 : read value of p

Step 12 : print the values of q, is rounded to p decimal places

and right justified in the field of w columns

Step 13 : print the value of q in exponential form with

default precision 6

Step 14 : print the value of r, is rounded to 2 decimal places and

left justified in the field of 7 columns

Step 15 : print the value of r in exponential form , is rounded

to 2 decimal places and right justified in the field

of 10 columns

Step 16 : read value of str

Step 17 : print the value of str in the field of 20 columns 3

Step 18 : print the first 10 characters of the string str in the field

of 30 columns and right justified

Step 19 : print the first 5 characters of the string str

Step 20 : print the first 10 characters of the string str in the field

of 20 columns and left justified

Step 21 : print the value of str

Step 22 : stop

PROGRAM

#include<stdio.h>

void main()

{

int a,b,d,w,p;

float q;

double r;

char str[20];

printf("FORMATTED INPUT & OUTPUT\n");

printf("Enter three integer numbers\n");

scanf("%d%\*d%d",&a,&b);

printf("value in a is%d\n Value in b is %d\n",a,b);

printf("Enter a 4 digit number\n");

scanf("%4d",&d);

printf("%6d\n",d);

printf("%2d\n",d);

printf("%-6d\n",d);

printf("%06d\n",d);

printf("Enter two floating point number\n");

scanf("%f%f",&q,&r);

printf("Enter the field width\n");

scanf("%d",&w);

printf("Enter the precision\n");

scanf("%d",&p);

printf("%\*.\*f\n",w,p,q);

printf("%e\n",q);

printf("%-7.2f\n",r);

printf("%10.2e\n",r);

printf("Enter a string\n");

scanf("%s",str);

printf("%20s\n",str);

printf("%30.10s\n",str);

printf("%.5s\n",str);

printf("%-20s\n",str);

printf("%5s\n",str);

}

OUTPUT

FORMATTED INPUT & OUTPUT

Enter three integer numbers

4

6

9

value in a is4

Value in b is 9

Enter a 4 digit number

2398

2398

2398

2398

002398

Enter two floating point number

3

1

Enter the field width

3

Enter the precision

2

3.00

3.000000e+00

0.00

5.26e-315

Enter a string

student

student

student

stude

student

student

PROGRAM NO :2

AIM : Write a program to reverse a number using typedef

ALGORITHM

Step1 : start

Step 2: declare int number as typedef

Step 3:declare n, temp, revrse=0 , r as number

Step 4:display enter a number to find reverse

Step 5: read n

Step 6:store n in temp

Step 7 : when temp > 0 goto step 8,9,10

Step 8: r=temp%10;

Step 9: revrse=(revrse\*10)+r;

Step 10: temp=temp/10;

Step 11: print reverse number

Step 12: stop

PROGRAM

#include<stdio.h>

main()

{

typedef int number;

number n,temp,revrse=0,r;

printf("\n Program to find reverse number");

printf("\nPlease enter any Number: ");

scanf("%d",&n);

temp=n;

while(temp>0)

{

r=temp%10;

revrse=(revrse\*10)+r;

temp=temp/10;

}

printf("\nReverse of %d is= %d",n,revrse);

return 0;

}

OUTPUT

Program to find reverse number

Please enter any Number: 4321

Reverse of 4321 is= 1234

PROGRAM NO: 3

AIM: Find grade of a student using else if ladder.

ALGORITHM

Step 1 : start

step 2:Display enter the mark of 5 subjects

Step 3 : read values of m1,m2, m3, m4 and m5 from user

Step 4 : check is each mark is greater than 30 goto step 5 other wise display failed

Step 5 : find total by adding each mark

Step 6 : find percentage by dividing by 5

Step 7 : Display total

Step 8 : Display percentage

step 9: when total greater than 450 display A grade

step 10: when total greater than 400 display B grade

step 10: when total greater than 350 display C grade

step 11: when total greater than 300 display D grade otherwise display E grade

step 12: stop

PROGRAM

#include<stdio.h>

main()

{

int m1,m2,m3,m4,m5,total,p;

printf("Enter the marks of 5 subjects:");

scanf("%d%d%d%d%d",&m1,&m2,&m3,&m4,&m5);

if((m1<30)||(m2<30)||(m3<30)||(m4<30)||(m4<30)||(m5<30))

{

printf("failed...");

}

else

{

total=m1+m2+m3+m4+m5;

p=(total/5);

printf("Total: %d\n",total);

printf("Percentage:%d\n",p);

if(total>450)

{

printf("A grade");

}

else if(total>400)

{

printf("B Grade");

}

else if(total>350)

{

printf("C Grade");

}

else if(total>300)

{

printf("D Grade");

}

else

{

printf("E Grade");

}

}

}

OUTPUT

Enter the marks of 5 subjects:78

90

67

66

50

Total: 351

Percentage:70

C Grade

PROGRAM NO :4

AIM : Program to print numbers equivalent to Roman Numbers using Switch case

ALGORITHM

Step 1 : start

Step 2 : declare ch as character

Step 3 : display enter the Roman Character

Step 4 : read ch

Step 5: select choice ch

Step 6 : if ch equal to I display 1

Step 7 : if ch equal to V display 5

Step 8 : if ch equal to X display 10

Step 9 : if ch equal to L display 50

Step 10 : if ch equal to C display 100

Step 11 : if ch equal to D display 500

Step 12 : if ch equal to F display 100

Step 13 : print “Invalid entry”

Step 14: stop

PROGRAM

#include<stdio.h>

main()

{

char ch;

printf("Enter Roman number \n");

scanf("%c",&ch);

switch(ch)

{

case 'I':

printf("1");

break;

case 'V':

printf("5");

break;

case 'X':

printf("10");

break;

case 'L':

printf("50");

break;

case 'C':

printf("100");

break;

case 'D':

printf("500");

break;

case 'F':

printf("1000");

break;

default:

printf("invalid entry");

}

}

OUTPUT

Enter Roman number

V

5

PROGRAM NO : 5

AIM : Program to print multiplication table using for loop

ALGORITHM

Step 1: start

Step 2: declare n ,i ,r as integers

Step 3: display enter the number

Step 4: read n

Step 5: display enter the range

Step 6: read r

Step 7: for i=0 to r goto step 8

Step 8: display i,n,i\*n

Step 9: stop

PROGRAM

#include<stdio.h>

main()

{

int n,i,r;

printf("Enter the number :");

scanf("%d",&n);

printf("Enter the range: ");

scanf("%d",&r);

for(i=1;i<=r;i++)

{

printf("%d \* %d = %d\n ",i,n,i\*n);

}

}

OUTPUT

Enter the number :7

Enter the range: 10

1 \* 7 = 7

2 \* 7 = 14

3 \* 7 = 21

4 \* 7 = 28

5 \* 7 = 35

6 \* 7 = 42

7 \* 7 = 49

8 \* 7 = 56

9 \* 7 = 63

10 \* 7 = 70

PROGRAM NO: 6

AIM: Find Armstrong numbers within a range using while loop.

ALGORITHM

Step 1 : start

Step 2 : read value of l from user

Step 3 : set i=l

Step 4 : Repeat while number less than or equal to l

Step 5 : set num=I and s=0

Step 6 : Repeat while num>0

Step 7 : calculate a=num%10

Step 8 : calculate p=d\*d\*d ,s=s+p and num=num/10.

Step 9 : check s==1 if it is true go to step 10

Step 10 : print value of i

Step 11 : increment I by 1 and go to step 4

Step 12 : stop

PROGRAM

#include<stdio.h>

void main()

{

int l,num,d,i;

printf("Enter a limit:\n");

scanf("%d",&l);

printf("Armstrong numbers are:\n");

i=1;

while(i<=l)

{

int p,s;

num=i;

s=0;

while(num>0)

{

d=num%10;

p=d\*d\*d;

s=s+p;

num=num/10;

}

if(s==i)

{

printf("%d\n",i);

}

i++;

}

}

OUTPUT

Enter a limit:

500

Armstrong numbers are:

1

153

370

371

407

PROGRAM NO : 7

AIM : Program to find the sum of first n natural numbers using do while loop

ALGORITHM

Step1 :Start

Step 2: display enter the number of natural numbers

Step 3: Read value of n

Step 4: set i=0

Step 5:print value of I and sum= sum+ i

Step 6: increment value of I by 1

Step 7 : Repeat step 5,6 until i < 1

Step 8 : print value of sum

Step 9 : Stop

PROGRAM

#include<stdio.h>

main()

{

int n,s=0,i;

printf("Enter the number of natural numbers :");

scanf("%d",&n);

i=0;

do

{

printf(" %d",i);

s=s+i;

i++;

}

while(i<=n);

printf("\nThe sum is :%d",s);

}

OUTPUT

Enter the number of natural numbers :8

0 1 2 3 4 5 6 7 8

The sum is :36

PROGRAM NO: 8

AIM: Implementing the break and continue statement.

ALGORITHM

Step 1 : start

Step 2 : read value of n from user

Step 3 : set i equal to 0

Step 4 : if i greater than n then go to step 5 else go to step 11

Step 5 : read value of a from user

Step 6 : if a equal to 9999 go to step 11 else go to step 7

Step 7 : if a greater than 0 go to step 10 else go to step 8

Step 8 : calculate s equal to sqrt(a)

Step 9 : print value of a and s

Step 10 : i equal to i+1 go to step 4

Step 11 : stop

PROGRAM

#include<stdio.h>

#include<math.h>

void main()

{

int n,a,i;

float s;

printf("Enter the limit\n");

scanf("%d",&n);

printf("Enter the numbers\n");

for(i=0;i<n;i++)

{

scanf("%d",&a);

if(a==9999)

{

break;

}

if(a<0)

{

continue;

}

s=sqrt(a);

printf("Square root of %d is %f\n",a,s);

}

}

OUTPUT

Enter the limit

3

Enter the numbers

4

Square root of 4 is 2.000000

9

Square root of 9 is 3.000000

16

Square root of 16 is 4.000000

PROGRAM NO: 9

AIM: Average of n numbers using goto statement.

ALGORITHM

Step 1 : start

Step 2 : read the value of n from user

Step 3 : set i equal to 0

Step 4 : if i greater than n then go to step 5 else go to step 9

Step 5 : read the value of p from user

Step 6 : if p less than 0 go to step 9 else go to step 7

Step 7 : calculate s equal to s+p

Step 8 : i equal to i + 1 go to step 4

Step 9 : calculate avg equal to (s/(i))

Step 10 : print the value of avg

Step 11 : stop

PROGRAM

#include<stdio.h>

main()

{

int n,p,i,s=0;

float avg;

printf("Enter limit\n");

scanf("%d",&n);

printf("Enter the elements\n");

for(i=0;i<n;i++)

{

scanf("%d",&p);

if(p<0)

{

goto label;

}

s=s+p;

}

label:

avg=(s/(i));

printf("Average=%f",avg);

}

OUTPUT

Enter limit

6

Enter the elements

2

6

23

17

45

33

Average=21.000000

PROGRAM NO:10

AIM: Call by value

ALGORITHM

Step 1 : start

Step 2 : Declare function sums

Step 3: declare s,n,m,sum=0 as integers

Step 4: Display Enter the number

Step 5: Read the number

Step 6:call function sums

Step 7 : print sum

Step 8 : stop

Function sums(int n)

Step 1:start

Step 2 : when n>0 goto step 3,4,5

Step 3 : m=n%10;

Step 4 : sum=sum+m;

Step 5 : n=n/10;

Step 6 : return sum

Step 7 : stop

PROGRAM

#include<stdio.h>

int sums(int);

int s, n,m,sum=0;

main()

{

printf("Enter the number:");

scanf("%d",&n);

s=sums(n);

printf("\n Sum:%d",s);

}

int sums(int n)

{

while(n>0)

{

m=n%10;

sum=sum+m;

n=n/10;

}

return sum;

}

OUTPUT

Enter the number:457

Sum:16

PROGRAM NO:11

AIM: Call by reference

ALGORITHM

Step 1 : start

Step 2 : read values of a and b from user

Step 3 : Display values before swapping

Step 4 : print a and b

Step 5 : call function swap(&a,&b)

step 6 : Display values after Swapping

step 7 : stop

Function swap(\*a,\*b)

Step 1 : start

Step 2 : set temp equal to \*a

Step 3 : set \*a equal to \*b

Step 4 : set \*b equal to temp

Step 5 : Stop

PROGRAM

#include<stdio.h>

void swap(int \*,int \*);

int a,b;

void main()

{

printf("Enter two values\n");

scanf("%d%d",&a,&b);

printf("Values before swapping\n");

printf("a=%d\tb=%d\n",a,b);

swap(&a,&b);

printf("Values after swapping\na=%d\tb=%d\n",a,b);

}

void swap(int \*a,int \*b)

{

int temp;

temp=\*a;

\*a=\*b;

\*b=temp;

}

OUTPUT

Enter two values

55

89

Values before swapping

a=55 b=89

Values after swapping

a=89 b=55

PROGRAM NO: 12

AIM: To find Fibonacci series by using recursive function.

ALGORITHM

Step 1 : start

Step 2 : set a=0,b=1 and t=3

Step 3 : read the value of n from user

Step 4 : Print values of a and b

Step 5: set f equal to call function fib(a,b)

Step 6 : stop

Function fact(n)

Step 1 : start

Step 2 : if t less than or equal to n then go to step 3

Step 3 : Find C=a+b

Step 4 : Print value of c

Step 5 : Increment value of t by 1

Step 6 : Call function fib(a,b)

Step 7 : return f

Step 8 : Stop

PROGRAM

#include<stdio.h>

int fib(int,int);

int t=3,n;

int main()

{

int a=0,b=1,f,c;

printf("Enter the Limit\n");

scanf("%d",&n);

printf("Fibonacci series\n");

printf("%d\n%d\n",a,b);

f=fib(a,b);

}

int fib(int a,int b)

{

int c,f;

if(t<=n)

{

c=a+b;

printf("%d\n",c);

t++;

f=fib(b,c);

return f;

}

}

OUTPUT:

Enter the Limit

10

Fibonacci series

0

1

1

2

3

5

8

13

21

34

PROGRAM NO: 13

AIM: Sum of elements in an array

ALGORITHM

Step 1: start

Step 2: read the value of n

Step 3: set i equal to 0

Step 4: if i less than n then go to step 5 else go to step 7

Step 5: read the value of a[i] from user

Step 6: i equal to i+1 go to step 4

Step 7: set s equal to call function fun()

Step 8: print the value of s

Step 9: stop

Function fun()

Step 1: start

Step 2: set i equal to 0 and sum=0

Step 3: if i less than n then go to step 4 else go to step 6

Step 4: calculate sum equal to sum+a[i]

Step 5: i equal to i+1 go to step 3

Step 6: return sum

Step 7: stop

PROGRAM

#include<stdio.h>

float fun();

int a[25],i,n;

void main()

{

int s;

printf("Enter a limit");

scanf("%d",&n);

printf("Enter numbers\n");

for(i=0;i<n;i++)

{

scanf("%d",&a[i]);

}

s=fun();

printf("Sum=%d",s);

}

float fun()

{

int sum=0;

for(i=0;i<n;i++)

{

sum=sum+a[i];

}

return sum;

}

Output

Enter a limit

5

Enter numbers

1

4

8

3

5

Sum=21

PROGRAM NO:14

AIM:Program to find minimum and maximum element of an array

ALGORITHM

Step 1: start

Step 2: read the value of limit from user

Step 3: set i equal to 0

Step 4: if i less than n then go to step 5 else go to step 7

Step 5: read the value of arr[i] from user

Step 6: i equal to i+1 got to step 4

Step 7: call function minmax()

Step 8: stop

Function minmax()

Step 1: start

Step 2: set large equal to 0

Step 3: set i equal to 0

Step 4: if i less than limit then go to step 5 else go to step 10

Step 5: if arr[i] greater than large step 6 else got to step 7

Step 6: set large equal to arr[i]

Step 7: if arr[i] less than small go to step 8 else got to step 9

Step 8: set s equal to arr[i]

Step 9: i equal to i+1 go to step 4

Step 10: print the value of large

Step 11: print the value of small

Step 12: stop

PROGRAM

include<stdio.h>

void minmax();

int arr[10],limit,i;

void main()

{

printf("Enter limit:\n");

scanf("%d",&limit);

printf("Enter numbers:\n");

for(i=0;i<limit;i++)

{

scanf("%d",&arr[i]);

}

minmax();

}

void minmax()

{

int large=0,small;

for(i=0;i<limit;i++)

{

if(arr[i]>large)

{

large=arr[i];

}

if(arr[i]<small)

{

small=arr[i];

}

}

printf("Largest number is=%d\n",large);

printf("Smallest number is=%d\n",small);

}

OUTPUT

Enter limit:

8

Enter numbers:

22

63

11

8

39

21

18

5

Largest number is=63

Smallest number is=5

PROGRAM NO: 15

AIM: Linear search using function.

ALGORITHM

Step 1: start

Step 2: read the value of num

Step 3: set i equal to 0

Step 4: if i less than num then go to step 5 else go to step 7

Step 5: read the value of a[i] from user

Step 6: i equal to i+1 got to step 4

Step 7: call function search ()

Step 8: if flag equal to 0 print element not found

Step 9: Stop

Function search()

Step 1: start

Step 2: read the value of s

Step 3: set i equal to 0

Step 4: if i less than num then go to step 5 else go to step 8

Step 5: if a[i] equal to s then go to step 6 else go to step 7

Step 6: print the value of i+1 go to step 10

Step 7: Set flag value equal to 1

Step 8: i equal to i+1 got to step 4

Step 9: Return value of flag

Step 10: stop

PROGRAM

#include<stdio.h>

int search();

int flag=0,num,a[25],p,i,l;

void main()

{

printf("Enter the limit\n");

scanf("%d",&num);

printf("Enter the numbers\n");

for(i=0;i<num;i++)

scanf("%d",&a[i]);

search();

if(flag==0)

printf("Not found\n");

}

int search()

{

int s;

printf("Enter the number to be searched\n");

scanf("%d",&s);

for(i=0;i<num;i++)

{

if(a[i]==s)

{

printf("Number found at %d position\n",i+1);

flag=1;

}

}

return flag;

}

OUTPUT

Enter the limit

8

Enter the numbers

2

1

6

9

2

4

6

3

Enter the number to be searched

3

Number found at 8 position

PROGRAM NO: 16

AIM: Bubble sort using function.

ALGORITHM

Step 1: start

Step 2: read the value of n from user

Step 3: set i equal to 0

Step 4: if i less than n then go to step 5 else go to step 7

Step 5: read the value of a[i] from user

Step 6: i equal to i+1 got to step 4

Step 7: call function bubble(a,n)

Step 8: stop

Function bubble(a[ ],n)

Step 1 : start

Step 2 : set i equal to 0

Step 3 : if i less than n then go to step 3 else go to step 12

Step 4 : set j equal to 0

Step 5 : if j less than n-i-1then go to step 6 else go to step 11

Step 6 : if ( a[j] > a[j+1] ) go to step 7 else got to step 10

Step 7 : set temp equal to a[j]

Step 8 : set a[j] equal to a[j+1]

Step 9 : set a[j+1] equal to temp

Step 10: j equal to j+1 go to step 5

Step 11: i equal to i+1 go to step 3

Step 12: set i equal to 0

Step 13: if i less than n then go to step 14 else go to step 16

Step 14: print the value of a[i]

Step 15: i equal to i+1 got to step 13

Step 16: stop

PROGRAM

#include<stdio.h>

void bubble(int [],int);

int i,j;

void main()

{

int a[10],n;

printf("Enter limit\n");

scanf("%d",&n);

printf("Enter numbers\n");

for(i=0;i<n;i++)

{

scanf("%d",&a[i]);

}

bubble(a,n);

}

void bubble(int a[],int n )

{

int temp;

for(i=0;i<n;i++)

{

for(j=0;j<(n-i-1);j++)

{

if(a[j]>a[j+1])

{

temp=a[j];

a[j]=a[j+1];

a[j+1]=temp;

}

}

}

printf("Sorted numbers are\n");

for(i=0;i<n;i++)

{

printf(" %d\n",a[i]);

}

}

output

Enter limit

6

Enter numbers

3

9

1

0

2

7

Sorted numbers are

0

1

2

3

7

9

PROGRAM NO: 17

AIM: Selection sort using function.

ALGORITHM

Step 1: start

Step 2: read the value of n from user

Step 3: set i equal to 0

Step 4: if i less than n then go to step 5 else go to step 7

Step 5: read the value of a[i] from user

Step 6: i equal to i+1 got to step 4

Step 7: call function selection (a,n)

Step 8: stop

Function selection(a[ ],n)

Step 1: start

Step 2: set i equal to 0

Step 3: if i less than n then go to step 3 else go to step 12

Step 4: set j equal to i+1

Step 5: if j less than n-1 then go to step 6 else go to step 11

Step 6: if ( a[i] > a[j] ) go to step 7 else got to step 10

Step 7: set temp equal to a[i]

Step 8: set a[i] equal to a[j]

Step 9: set a[j] equal to temp

Step 10: j equal to j+1 go to step 5

Step 11: i equal to i+1 go to step 3

Step 12: set i equal to 0

Step 13: if i less than n then go to step 14 else go to step 16

Step 14: print the value of a[i]

Step 15: i equal to i+1 got to step 13

Step 16: stop

PROGRAM

#include<stdio.h>

void selection(int [],int);

int i,j;

void main()

{

int a[10],n;

printf("Enter limit\n");

scanf("%d",&n);

printf("Enter numbers\n");

for(i=0;i<n;i++)

{

scanf("%d",&a[i]);

}

selection(a,n);

}

void selection(int a[],int n)

{

int temp;

for(i=0;i<n;i++)

{ for(j=i+1;j<n;j++)

{ if(a[i]>a[j])

{

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

printf("Sorted numbers are\n");

for(i=0;i<n;i++)

{

printf(" %d\n",a[i]);

}

}

OUTPUT:

Enter limit

7

Enter numbers

1

4

9

0

4

6

5

Sorted numbers are

0

1

4

4

5

6

9

PROGRAM NO: 18

AIM: Matrix multiplication using function.

ALGORITHM

Step 1 : start

Step 2 : read the value of r1 and c1

Step 3 : read the value of r2 and c2 from user

Step 4 : if r1 not equal to c2 then print multiplication not possible else go to step 5

Step 5 : set i equal to 0

Step 6 : if i less than r2 then go to step 7 else go to step 12

Step 7 : set j equal to 0

Step 8 : if j less than c2 then go to step 9 else go to step 11

Step 9 : read the value of a[i][j] from user

Step 10: j equal to j+1 got to step 8

Step 11: i equal to i+1 go to step 6

Step 12: set i equal to 0

Step 13: if i less than r1 then go to step 14 else go to step 19

Step 14: set j equal to 0

Step 15: if j less than c2 then go to step 16 else go to step 18

Step 16: read the value of b[i][j] from user

Step 17: j equal to j+1 got to step 15

Step 18: i equal to i+1 go to step 13

Step 19: Print two metrices a[i][j] and b[i][j]

Step 20: Call function mul(a,b,r1,c2)

Step 21: stop

Function mul()

Step 1 : start

Step 2 : set i equal to 0

Step 3 : if i less than r1 then go to step 4 else go to step 13

Step 4 : set j equal to 0

Step 5 : if j less than c2 then go to step 6 else go to step 8

Step 6 : set m[i][j] equal to 0

Step 7 : set k equal to 0

Step 8 : if k less than r1 then go to step 9 else go to step 11

Step 9 : calculate m[i][j] equal to m[i][j]+(a[i][k]\*b[k][j])

Step 10: k equal to k+1 go to step 8

Step 11: j equal to j+1 go to step 5

Step 12: i equal to i+1 go to step 3

Step 13: set i equal to 0

Step 14: if i less than m then go to step 15 else go to step 20

Step 15: set j equal to 0

Step 16: if j less than n then go to step 17 else go to step 19

Step 17: print the value of m[i][j]

Step 18: j equal to j+1 go to step 16

Step 19: i equal to i+1 go to step 14

Step 20: stop

PROGRAM

#include<stdio.h>

void mul(int [][10],int [][10],int ,int);

int i,j,r1,c1,r2,c2,k;

void main()

{

int a[10][10],b[10][10];

printf("Enter row and column of first matrix\n");

scanf("%d%d",&r1,&c1);

printf("Enter row and column of second matrix\n");

scanf("%d%d",&r2,&c2);

if(c1!=r2)

printf("not possible\n");

else

{

printf("Enter the elements of first matrix\n");

for(i=0;i<r1;i++)

{

for(j=0;j<c1;j++)

{

scanf("%d",&a[i][j]);

}

}

printf("Enter the elements of second matrix\n");

for(i=0;i<r2;i++)

{

for(j=0;j<c2;j++)

{

scanf("%d",&b[i][j]);

}

}

printf("FIRST MATRIX\n");

for(i=0;i<r1;i++)

{

for(j=0;j<c1;j++)

{

printf("%d\t",a[i][j]);

}

printf("\n");

}

printf("SECOND MATRIX\n");

for(i=0;i<r2;i++)

{

for(j=0;j<c2;j++)

{

printf("%d\t",b[i][j]);

}

printf("\n");

}

mul(a,b,r1,c2);

}

}

void mul(int a[][10],int b[][10],int r1,int c2)

{

int m[10][10];

printf("PRODUCT MATRIX\n");

for(i=0;i<r1;i++)

{

for(j=0;j<c2;j++)

{

m[i][j]=0;

for(k=0;k<r1;k++)

{

m[i][j]=m[i][j]+(a[i][k]\*b[k][j]);

}

printf("%d\t",m[i][j]);

}

printf("\n");

}

}

OUTPUT

Enter row and column of first matrix

2

2

Enter row and column of second matrix

2

2

Enter the elements of first matrix

2

1

3

4

Enter the elements of second matrix

1

2

3

4

FIRST MATRIX

2 1

3 4

SECOND MATRIX

1 2

3 4

PRODUCT MATRIX

5 8

15 22

PROGRAM NO :19

Aim : program to find trace of a matrix

ALGORITHM

Step 1: start

Step 2:declare a[10][10],I,j,sum=0,m,n as integers

Step 3: display enter the values of m and n

Step 4:read m and n

Step 5:display enter the elements of the matrix

Step 6:for i=0 to m and j=0 to n read the elements of the matrix

Step 7: for i=0 to m and j=0 to n goto step 8

Step 8: if i = j add array elements to sum

Step 9: display sum

Step 10: stop

PROGRAM

#include<stdio.h>

void dioganal(int [][10],int,int);

int i,j;

void main()

{

int a[10][10],r,c;

printf("Enter row and column of matrix\n");

scanf("%d%d",&r,&c);

if(r!=c)

printf("Not possible\n");

else

printf("Enter the elements of matrix\n");

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

{

for(j=0;j<c;j++)

{

scanf("%d",&a[i][j]);

}

}

printf("Matrix is\n");

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

{

for(j=0;j<c;j++)

{

printf("%d\t",a[i][j]);

}

printf("\n");

}

dioganal(a,r,c);

}

void dioganal(int a[][10],int r,int c)

{

int sum=0;

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

{

for(j=0;j<c;j++)

{

if(i==j)

{

sum=sum+a[i][j];

}

}

}

printf("\nTrace of matrix is=%d",sum);

}

OUTPUT

Enter row and column of matrix

3

3

enter the elements of matrix

1

3

4

5

2

3

5

7

1

Matrix is

1 3 4

5 2 3

5 7 1

Trace of Matrix is=4

PROGRAM NO: 20

AIM: String manipulations.

ALGORITHM

Step 1 : start

Step 2 : read the value of s from user

Step 3 : find the length of string s then go to step 4

Step 4 : print the string length

Step 5 : read value of s1 and s2 from user

Step 6 : copy the string s2 to s1

Step 7 :print the value of s1

Step 8 : read value of s3 and s4from user

Step 9 : compare s3 and s4 if it is equal to 0 then go to step 10 else go to step 11

Step 10 : print strings are equal

Step 11 : print strings are not equal

Step 12 : concatenate strings s3 and s4

Step 13 : print value of s3

Step 14: stop

PROGRAM

#include<stdio.h>

#include<string.h>

main()

{

char s[20],s1[10],s2[10],s3[10],s4[10];

int n,h;

printf("Enter the string:\n");

scanf("%s",s);

n=strlen(s);

printf(“Finding Length\n");

printf("Length of string using strlen: %d\n",n);

printf("Enter two strings:\n");

scanf("%s%s",s1,s2);

strcpy(s1,s2);

printf("\nCopying String\n");

printf("Copied string is:%s\n",s1);

printf("Enter two strings:\n");

scanf("%s%s",s3,s4);

printf("Comparing Strings\n");

h=strcmp(s3,s4);

if(h==0)

{

printf("Two strings are equal\n");

}

else

{

printf("Two strings are not equal\n");

}

strcat(s3,s4);

printf("Concatenating Strings\n");

printf("The concatenated string:%s\n",s3);

}

OUTPUT:

Enter the string:

Greetings

Finding Length

Length of string using strlen: 9

Enter two strings:

smile

please

Copying String

Copied string is:please

Enter two strings:

take

care

Comparing Strings

Two strings are not equal

Concatenating Strings

The concatenated string:takecare

PROGRAM NO:21

AIM: implementing simple pointer

ALGORITHM

step 1: start

step 2: declare interger variable i equal to 0 and pointer variable \*p

step 3: declare floating point variable f equal to 3.4 and pointer variable

\*fp

step 4: declare char variable c equal to 'a' and a pointer \*cp

step 5: print i,f,c

step 6: ip equal to &f

step 7: print value of ip and \*ip

step 8: fp equal to &f

step 9: print values of fp and \*fp

step 10: cp equal to &c

step 11: print values of cp and \*cp

step 12: stop

PROGRAM

#include<stdio.h>

main()

{

int i=21,\*ip;

float f=33.9,\*fp;

char c='j',\*cp;

printf("i=%d\n",i);

printf("f=%f\n",f);

printf("c=%c\n",c);

ip=&i;

printf("Values of i=%u\n",ip);

printf("Values of i=%d\n",\*ip);

fp=&f;

printf("Address of f=%u\n",fp);

printf("Values of f=%f\n",\*fp);

cp=&c;

printf("Address of c=%u\n",cp);

printf("Values of c=%c\n",\*cp);

}

OUTPUT

i=21

f=33.900002

c=j

Values of i=3225988948

Values of i=21

Address of f=3225988944

Values of f=33.900002

Address of c=3225988943

Values of c=j

PROGRAM NO :22

AIM :array of pointers

Step 1 : start

Step 2 : read the value of n from user

Step 3 : set i equal to 0

Step 4 : if i less than n then go to step 5 else go to step 10

Step 5 : read the value of s[i] from user

Step 6 : i equal to i+1 got to step 4

Step 7 : set i equal to 0

Step 8 : if i less than n then go to step 9 else go to step 16

Step 9 : set j equal to i+1

Step 10: if j less than n then go to step 11 else go to step 15

Step 11: compare &s[i ]and &s[j] if it is greater than 0 then go to step 12 else go to step15

Step 12: set temp equal to s[i]

Step 13: set s[i] equal to s[j]

Step 14: set s[j] equal to temp

Step 15: j equal to j+1 go to step 10

Step 16: i equal to i+1 go to step 8

Step 17: print the value of s[i]

Step 18: stop

PROGRAM

#include<stdio.h>

void main()

{

char \*s[20],\*temp;

int i,j,n;

printf("Enter limits:\n");

scanf("%d",&n);

printf("Enter names:\n");

for(i=0;i<n;i++)

{

scanf("%s",&s[i]);

}

for(i=0;i<n;i++)

{

for(j=i+1;j<n;j++)

{

if(strcmp(&s[i],&s[j])>0)

{

temp=s[i];

s[i]=s[j];

s[j]=temp;

}

}

}

printf("Sorted names are:\n");

for(i=0;i<n;i++)

{

printf("%s\n",&s[i]);

}

}

OUTPUT

Enter limits:

4

Enter names:

Deepak

Rijil

Arya

Qweeza

Sorted names are:

Arya

Deepak

Qweeza

Rijil

PROGRAM NO: 23

AIM: Implementation of pointer to pointer

ALGORITHM

Step 1 : start

Step 2 : set p1 equal to &a

Step 3 : set p2 equal to &p1

Step 4 : set q1 equal to &b

Step 5 : set q2 equal to &q1

Step 7 : read the value of a and b

Step 8 : calculate c equal to \*\*p2 + \*\*q2

Step 9 : print the value of c

Step 10 : calculate c equal to \*\*p2 - \*\*q2

Step 11 : print the value of c

Step 12 : calculate c equal to \*\*p2 \* \*\*q2

Step 13 : print the value of c

Step 14 : calculate c equal to \*\*p2 / \*\*q2

Step 15 : print the value of c

Step 16 : stop

PROGRAM

#include<stdio.h>

void main()

{

int a,b,c,\*p1,\*\*p2,\*q1,\*\*q2;

p1=&a;

p2=&p1;

q1=&b;

q2=&q1;

printf("Enter two numbers\n");

scanf("%d%d",&a,&b);

c=\*\*p2+\*\*q2;

printf("Sum=%d\n",c);

c=\*\*p2-\*\*q2;

printf("Difference=%d\n",c);

c=\*\*p2\*\*\*q2;

printf("Product=%d\n",c);

c=\*\*p2/ \*\*q2;

printf("Division=%d\n",c);

}

OUTPUT

Enter two numbers

2

2

Sum=4

Difference=0

Product=4

Division=1

PROGRAM NO: 24

AIM:Employee structure using function.

ALGORITHM

Step 1 : start

Step 2 : create a structure emp

Step 3 : create structure variable e[10]

Step 4 : print menu and select choice s

Step 5 : read the value of ch from user

Step 6 : if s equal to 1 call function read() and go to step 12

Step 7 : if s equal to 2 call function display() and go to step 12

Step 8 : if s equal to 3 call function search() and go to step 12

Step 9 : if s equal to 4 call function del() and go to step 12

Step 10 : if s equal to 5 go to step 12 else go to step 11

Step 11 : print â€œInvalidâ€

Step 12 : if s less than or equal to 4 go to step 3 else go to step 13

Step 13 : stop

Function read()

Step 1 : start

Step 2 : read the value of n from user

Step 3 : set i equal to 0

Step 4 : if i less than n then go to step 5 else go to step 11

Step 5 : print the value of i+1

Step 6 : read the value of e[i].empnofrom user

Step 7 : read the value of e[i].name from user

Step 8 : read the value of e[i].salaryfrom user

Step 9 : i equal to i+1 go to step 4

Step 10 : stop

Function display()

Step 1 : start

Step 2 : set i equal to 0

Step 3 : if i less than n then go to step 4 else go to step 6

Step 4 : print values of e[i].empno,e[i].name,e[i].salary

Step 5 : i equal to i+1 go to step 3

Step 6 : stop

Function search()

Step 1 : start

Step 2 : set f equal to 0

Step 3 : read the value of m from user

Step 4 : set i equal to 0

Step 5 : if i less than n then go to step 6 else go to step 13

Step 6 : if (e[i].name,m==k) go to step 7 else go to step12

Step 7 : print the value of e[i].empno

Step 8 : print the value of e[i].name

Step 9 : print the value of e[i].salary

Step 10: i equal to i+1 go to step 5

Step 11 : if I equal to n goto step 12

Step 12 : print â€œ not foundâ€

Step 13 : stop

Function sort()

PROGRAM

#include<stdio.h>

int n,i,s,c;

char m[10],k[10];

struct employee

{

int empno;

char name[20];

float salary;

}e[10];

void read(struct employee[]);

void display(struct employee[]);

void search(struct employee[]);

void sort(struct employee[],int);

main()

{

do

{

printf("\n 1 ENTER");

printf("\n 2 DISPLAY");

printf("\n 3 SEARCH");

printf("\n 4 SORTED ");

printf("\nEnter your choice:\n");

scanf("%d",&s);

switch(s)

{

case 1:

read(e);

break;

case 2:

display(e);

break;

case 3:

search(e);

break;

case 4:

sort(e,n);

break;

default:

printf("invalid\n");

}

printf("\nDo you want to continue \n1 - CONTINUE\n0 - EXIT\t");

scanf("%d",&c);

}

while(c==1);

}

void read(struct employee e[])

{

printf("Enter limits:\n");

scanf("%d",&n);

printf("Enter Employee Details:\n");

for(i=0;i<n;i++)

{

printf("Enter Employee-Id:");

scanf("%d",&e[i].empno);

printf("Enter Name:");

scanf("%s",e[i].name);

printf("Enter Salary:");

scanf("%f",&e[i].salary);

}}

void display(struct employee e[])

{

printf("\nEmployee Details:\n");

for(i=0;i<n;i++)

{

printf("Employee-Id:%d\n",e[i].empno);

printf("Name:%s\n",e[i].name);

printf("Salary:%f\n",e[i].salary);

}

}

void sort(struct employee e[],int n)

{

int j,m;

struct employee temp;

printf("\nthe sorted list of names are:");

for(m=0;m<=n-1;m++)

{

for(j=m+1;j<n;j++)

{

if(strcmp(e[m].name,e[j].name)>0)

{

temp=e[m];

e[m]=e[j];

e[j]=temp;

}

}

}

display(e);

}

void search(struct employee e[])

{

printf("Enter the name to be searched:\n");

scanf("%s",m);

for(i=0;i<n;i++)

{

if(strcmp(e[i].name,m)==0)

{

printf("Name found\n");

printf("Employee-Id:%d\n",e[i].empno);

printf("Name:%s\n",e[i].name);

printf("Salary:%f\n",e[i].salary);

break;

}

}

if(i==n)

printf("Name not found\n");

else

{

printf("%d\n%s\n%f\n",e[i].empno,e[i].name,e[i].salary);

}}

OUTPUT

1 ENTER

2 DISPLAY

3 SEARCH

4 SORTED

Enter your choice:

1

Enter limits:

3

Enter Employee Details:

Enter Employee-Id:101

Enter Name:Sabu

Enter Salary:38000

Enter Employee-Id:102

Enter Name:Nitheesh

Enter Salary:5600

Enter Employee-Id:103

Enter Name:Sree

Enter Salary:71000

Do you want to continue

1 - CONTINUE

0 - EXIT 1

1 ENTER

2 DISPLAY

3 SEARCH

4 SORTED

Enter your choice:

2

Employee Details:

Employee-Id:101

Name:Sabu

Salary:38000.000000

Employee-Id:102

Name:Nitheesh

Salary:5600.000000

Employee-Id:103

Name:Sree

Salary:71000.000000

Do you want to continue

1 - CONTINUE

0 - EXIT 1

1 ENTER

2 DISPLAY

3 SEARCH

4 SORTED

enter your choice:

3

Enter the name to be searched:

Sree

Name found

Employee-Id:103

Name:Sree

Salary:71000.000000

103

Sree

71000.000000

Do you want to continue

1 - CONTINUE

0 - EXIT 1

1 ENTER

2 DISPLAY

3 SEARCH

4 SORTED

enter your choice:

4

the sorted list of names are:

Employee Details:

Employee-Id:102

Name:Nitheesh

Salary:5600.000000

Employee-Id:101

Name:Sabu

Salary:38000.000000

Employee-Id:103

Name:Sree

Salary:71000.000000

Do you want to continue

1 - CONTINUE

0 - EXIT 0

PROGRAM 25

AIM:Illustrate the concept of union using function.

ALGORITHM

1:Start

2:Read roll no,name,mark

3:Define student record using union

4:Create a student union variable

5:Call student of function

6:stop

7:Student function

8:Start

9:Read the value for each element of the student record

10:Display each element of the student record

PROGRAM

#include<stdio.h>

union student

{

int rollno;

char name[20];

int mark;

}s;

void student();

main()

{

student();

}

void student()

{

printf("Enter roll no \n");

scanf("%d",&s.rollno);

printf("Rollno is %d \n",s.rollno);

printf("Enter name \n");

scanf("%s",s.name);

printf("Name is %s \n",s.name);

printf("Enter mark\n");

scanf("%d",&s.mark);

printf("mark is %d \n",s.mark);

}

OUTPUT

Enter roll no

101

Rollno is 101

Enter name

Priya

Name is Priya

Enter mark

45

mark is 45

PROGRAM NO: 26

AIM: Illustrate enumarator datatype

ALGORITHM

Step 1 : start

Step 2 : create enumerated data type subjects

Step 3 : create enumerated variable sub

Step 4 :set sub equal to Malayalam

Step 5 : if sum less than or equal to mathematics then go to step 6 else goto7

Step 6 : read value of marks from user

Step 7 : sub equal to sub+1

Step 8: print values of marks of each subject

Step 9: stop

PROGRAM

#include<stdio.h>

enum subjects

{

malayalam,

english,

physics,

maths,

computerscience

};

main()

{

int marks[5];

enum subjects subject;

printf("Enter marks in five subjects\n");

for(subject=malayalam;subject<=computerscience;subject++)

scanf("%d",&marks[subject]);

printf("marks in five different subjects\n");

printf("malayalm=%d\n", marks[malayalam]);

printf("english=%d\n" ,marks[english]);

printf("physics=%d\n" ,marks[physics]);

printf("maths=%d\n" ,marks[maths]);

printf("computerscience=%d\n", marks[computerscience]);

}

OUTPUT

Enter marks in five subjects

67

89

90

55

73

marks in five different subjects

malayalm=67

english=89

physics=90

maths=55

computerscience=73

PROGRAM NO:27

AIM: Illustrate the concept of bitfield.

ALGORITHM

Step 1 : start

Step 2 : create a structure sample

Step 3 : create structure variable e

Step 4 : set e.a equal to 4

Step 5 : set e.b equal to 1

Step 6 : print the value of e.a

Step 7 : print the value of e.b

Step 8 : calculate c equal to e.a + e.b

Step 9 : print the value ofc

Step 10 : calculate c equal to e.a - e.b

Step 11 : print the value of c

Step 12 : calculate c equal to e.a \* e.b

Step 13 : print the value of c

Step 14 : stop

PROGRAM

#include<stdio.h>

struct temp

{

unsigned int a:3;

unsigned int b:3;

};

main()

{

struct temp t;

int c;

t.a=6;

t.b=3;

printf("%d \t",t.a);

printf("%d \t",t.b);

c=t.a+t.b;

printf("Sum=%d\n",c);

c=t.a-t.b;

printf("Difference=%d\n",c);

c=t.a\*t.b;

printf("Product=%d\n",c);

c=t.a/t.b;

printf("Division=%d\n",c);

}

OUTPUT

6 3

Sum=9

Difference=3

Product=18

Division=2