Exercise-Input-Output in C

1.

#include<stdio.h>

#define MSSG "Hello World\n"

int main(void)

{

printf(MSSG);

return 0;

}

2.

#include<stdio.h>

int main(void)

{

printf("Indian\b \n");

printf("New\rDelhi\n");

return 0;

}

3.

#include<stdio.h>

int main(void)

{

int a=11;

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

printf("a=%o\t",a);

printf("a=%x\t",a);

printf("a=%X\n",a);

return 0;

}

4.

#include<stdio.h>

#include<limits.h>

int main(void)

{

int a=4000000000;

unsigned int b=4000000000;

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

printf("%d, %u\n",INT\_MAX,UINT\_MAX);

return 0;

}

5.

#include<stdio.h>

int main(void)

{

char ch;

printf("Enter a character : ");

scanf("%c",&ch);

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

return 0;

}

6.

#include<stdio.h>

int main(void)

{

float b=123.1265;

printf("%f\t",b);

printf("%.2f\t",b);

printf("%.3f\n",b);

return 0;

}

7.

#include<stdio.h>

int main(void)

{

int a=625,b=2394,c=12345;

printf("%5d,%5d,%5d\n",a,b,c);

printf("%3d,%4d,%5d\n",a,b,c);

return 0;

}

8.

#include<stdio.h>

int main(void)

{

int a=98;

char ch='c';

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

return 0;

}

9.

#include<stdio.h>

int main(void)

{

float a1,b1,a2,b2,a3,b3;

a1=2;

b1=6.8;

a2=4.2;

b2=3.57;

a3=9.82;

b3=85.673;

printf("%3.1f,%4.2f\n",a1,b1);

printf("%5.1f,%6.2f\n",a2,b2);

printf("%7.1f,%8.2f\n",a3,b3);

return 0;

}

10.

#include<stdio.h>

int main(void)

{

printf("%10s\n","India");

printf("%4s\n","India");

printf("%.2s\n","India");

printf("%5.2s\n","India");

return 0;

}

Exercise- Operations and Expressions

1.

#include<stdio.h>

int main(void)

{

int a=-3;

a = -a-a+!a;

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

return 0;

}

2.

#include<stdio.h>

int main(void)

{

int a=2,b=1,c,d;

c = a<b;

d = (a>b) && ( c<b);

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

return 0;

}

3.

#include<stdio.h>

int main(void)

{

int a=9,b=15,c=16,d=12,e,f;

e = !(a<b || b<c);

f = (a>b) ? a-b: b-a;

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

return 0;

}

4.

#include<stdio.h>

int main(void)

{

int a=5;

a=6;

a=a+5\*a;

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

return 0;

}

5.

#include<stdio.h>

int main(void)

{

int a=5, b=5;

printf("%d,%d\t",++a,b--);

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

printf("%d,%d\t",++a,b++);

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

return 0;

}

6.

#include<stdio.h>

int main(void)

{

int x,y,z;

x=8++;

y=++x++;

z=(x+y)--;

printf("x=%d, y=%d, z=%d\n",x,y);

return 0;

}

7.

#include<stdio.h>

int main(void)

{

int a=4, b=8, c=3, d=9, z;

z = a++ + ++b \* c-- - --d;

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

return 0;

}

8.

#include<stdio.h>

int main(void)

{

int a=14,b,c;

a=a%5;

b=a/3;

c=a/5%3;

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

return 0;

}

9.

#include<stdio.h>

int main(void)

{

int a=15,b=13,c=16,x,y;

x = a-3%2+c\*2/4%2+b/4;

y = a = b+5-b+9/3;

printf("x=%d, y=%d\n",x,y);

return 0;

}

10.

#include<stdio.h>

int main(void)

{

int x,y,z,k=10;

k+=(x=5, y=x+2, z=x+y);

printf("x=%d, y=%d, z=%d, k=%d\n",x,y,z,k);

return 0;

}

11.

#include<stdio.h>

int main(void)

{

float b;

b = 15/2;

printf("%f\t",b);

b = (float)15/2 + (15/2);

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

return 0;

}

12.

#include<stdio.h>

int main(void)

{

int a=9;

char ch='A';

a=a+ch+24;

printf("%d,%c\t%d,%c\n",ch,ch,a,a);

return 0;

}

13.

#include<stdio.h>

int main(void)

{

int a,b,c,d;

a=b=c=d=4;

a\*=b+1;

c+=d\*=3;

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

return 0;

}

14.

#include<stdio.h>

int main(void)

{

int a=5,b=10,temp;

temp=a,a=b,b=temp;

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

return 0;

}

15.

#include<stdio.h>

int main(void)

{

int a=10, b=3, max;

a>b ? max=a : max=b;

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

return 0;

}

16.

#include<stdio.h>

int main(void)

{

int a=5, b=6;

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

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

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

return 0;

}

17.

#include<stdio.h>

int main(void)

{

int a=3,b=4,c=3,d=4,x,y;

x = (a=5) && (b=7);

y = (c=5) || (d=8);

printf("a=%d, b=%d, c=%d, d=%d, x=%d, y=%d\n",a,b,c,d,x,y);

x = (a==6) && (b=9);

y = (c==6) || (d=10);

printf("a=%d, b=%d, c=%d, d=%d, x=%d, y=%d\n",a,b,c,d,x,y);

return 0;

}

18.

#include<stdio.h>

int main(void)

{

int a=10;

a=a++;

a = a++ \* a--;

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

printf("%d\n",a++ \* a++);

return 0;

}

19.

#include<stdio.h>

int main(void)

{

int a=2,b=2,x,y;

x = 4\*(++a \* 2 + 3);

y = 4\*(b++ \* 2 + 3 );

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

return 0;

}

20.

#include<stdio.h>

int main(void)

{

float tempC,tempF;

printf("Enter the temprature in Celsius : ");

scanf("%f",&tempC);

tempF = (tempC \* 9/5) + 32;

printf("Temprature in Fahrenheit is : %f\n", tempF);

return 0;

}

21.

#include<stdio.h>

int main(void)

{

float r,area,perimeter;

printf("Enter radius of the circle : ");

scanf("%f",&r);

area = 22.0/7 \* r \* r;

perimeter = 2 \* 22.0/7 \* r;

printf("Area=%f, Perimeter=%f\n",area,perimeter);

return 0;

}

22.

#include<stdio.h>

int main(void)

{

int n;

printf("Enter a number : ");

scanf("%d",&n);

printf("Octal - %o, Hexadecimal - %x\n",n,n);

return 0;

}

23.

#include<stdio.h>

int main(void)

{

int n;

printf("Enter a number : ");

scanf("%d",&n);

printf("Remainder = %d\n",n%3);

return 0;

}

24.

#include<stdio.h>

int main(void)

{

int n;

printf("Enter a number : ");

scanf("%d",&n);

printf("Remainder = %d\n",n%3);

return 0;

}

25.

#include<stdio.h>

int main(void)

{

float m1,m2,m3,m4,m5,per;

printf("Enter marks : ");

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

per = (m1+m2+m3+m4+m5)/500 \* 100;

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

return 0;

}

Exercise-Control Statements

1.

#include<stdio.h>

int main(void)

{

int a=9;

if(a=5)

printf("a is five\t");

else

printf("a is not five\t");

printf("Value of a is %d\n",a);

return 0;

}

2.

#include<stdio.h>

int main(void)

{

int a=0;

if(a=0)

printf("a is zero\t");

else

printf("a is not zero\t");

printf("Value of a is %d\n",a);

return 0;

}

3.

#include<stdio.h>

int main(void)

{

int i=10;

i==50;

if(i==50)

printf("i is fifty\n");

else

printf("i is not fifty\n");

return 0;

}

4.

#include<stdio.h>

int main(void)

{

int a=20,b=3;

if(a<10)

a=a-5;

b=b+5;

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

return 0;

}

5.

#include<stdio.h>

int main(void)

{

int a=9,b=0,c=0;

if(!a<10 && !b||c)

printf("C in depth\n");

else

printf("See in depth\n");

return 0;

}

6.

#include<stdio.h>

int main(void)

{

int i=1, j=9;

if(i>=5 && j<5);

i = j+2;

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

return 0;

}

7.

#include<stdio.h>

int main(void)

{

int a=0,b=0;

if(!a)

{

b = !a;

if(b)

a = !b;

}

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

return 0;

}

8.

#include<stdio.h>

int main(void)

{

int a=2,x=10;

if(a==2)

if(x==8)

printf("a is 2 and x is 8\n");

else

printf("a is not 2\n");

return 0;

}

9.

#include<stdio.h>

int main(void)

{

int x=10, a=15, b=6;

x+= a<b ? (-x) : 100;

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

return 0;

}

10.

#include<stdio.h>

int main(void)

{

int i=3,j=2,k=2,l=10,m=1,z=0;

if(i==3)

if(j==k)

if(l<m)

z=100;

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

return 0;

}

11.

#include<stdio.h>

int main(void)

{

int x=2,y=0;

if(x==0)

y++;

if(x>0)

y--;

if(x<0)

y+=2;

printf("%d %d\n",x,y);

return 0;

}

12.

#include<stdio.h>

int main(void)

{

char grade='A';

int marks = 94;

if(grade == 'A')

if(marks > 95)

printf("Excellent\n");

else

printf("Work hard for getting A grade\n");

return 0;

}

13.

#include<stdio.h>

int main(void)

{

int a=10,b=80,c=30;

if(a==10)

if(b==20)

if(c==30)

printf("a is 10,b is 20, c is 30\n");

else

printf("a is 10, b is not 20\n");

else

printf("a is not 10\n");

return 0;

}

14.

#include<stdio.h>

int main(void)

{

int k=10;

switch(k)

{

case '5':

case '10':

k++;

continue;

case '15':

case '20':

k--;

}

return 0;

}

15.

#include<stdio.h>

int main(void)

{

int var=2,x=1,y=2;

switch(var)

{

case x:

x++;

break;

case y:

y++;

break;

}

return 0;

}

16.

#include<stdio.h>

int main(void)

{

int i,total=0;

for(i=1; i<=10; i++)

{

switch(i)

{

case 1:

case 4:

case 5:

case 7:

total+=i;

break;

default:

continue;

}

printf("%d ",i);

}

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

return 0;

}

17.

#include<stdio.h>

int main(void)

{

int x=2,y=20;

switch(x)

{

y=30;

case 1:

y++;

break;

case 2:

y--;

break;

default:

y=y+2;

}

printf("y is %d\n",y);

return 0;

}

18.

#include<stdio.h>

int main(void)

{

int x,y;

x=3,y=4;

if(x==1)

y=x+1;

else if(x==2)

{

y=0;

x=0;

}

else if(x==3 || x==4 || x==5)

y++;

else if(x==6)

y+=4;

else

y--;

printf("%d %d\n", x,y);

x=6,y=4;

if(x==1)

y=x+1;

else if(x==2 || x==3|| x==4)

y++;

else if(x==5)

y--;

else if(x==6)

y=0;

printf("%d %d\n", x,y);

return 0;

}

19.

#include<stdio.h>

int main(void)

{

int x=30,y=5,z=10;

switch(x)

{

case 10:

z++;

break;

case 20:

z--;

break;

case 30:

switch(y)

{

case 2:

z=z+3;

break;

case 4:

case 5:

z=z+10;

break;

default:

z=z-2;

}

break;

default:

z=0;

}

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

return 0;

}

20.

#include<stdio.h>

int main(void)

{

int a=5;

begin:

if(a)

{

printf("%d ",a);

a--;

goto begin;

}

return 0;

}

21.

#include<stdio.h>

int main(void)

{

int a,b;

while(-1)

{

printf("Test Loop\n");

}

22.

#include<stdio.h>

int main(void)

{

int i,j;

/\*(i)\*/

for(i=0; i<10; i++)

printf("%d ",i);

/\*(ii)\*/

/\* for(i=1; i<=10; i++)

printf("%d ",i);

\*/

/\*(iii)\*/

/\* for(i=0; i<=10; i++)

printf("%d ",i);

\*/

/\*(iv)\*/

/\* for(i=1; i<10; i++)

printf("%d ",i);

\*/

/\*(v)\*/

/\* for(i=0; i<=10; i--)

printf("%d ",i);

\*/

/\*(vi)\*/

/\* for(i=10; i>=1; i--)

printf("%d ",i);

\*/

/\*(vii)\*/

/\* for(i=10; i>1; i--)

printf("%d ",i);

\*/

/\*(viii)\*/

/\* for(i=10; i>0; i--)

printf("%d ",i);

\*/

/\*(ix)\*/

/\* for(i=15; i>=0; i=i-3)

printf("%d ",i);

\*/

/\*(x)\*/

/\* for(i=10; i>=0; i--);

printf("%d ",i);

\*/

/\*(xi)\*/

/\* for(i=0; i>10; i++)

printf("%d ",i);

\*/

/\*(xii)\*/

/\* for(i=0; i<=10; i+=20)

printf("%d ",i);

\*/

/\*(xiii)\*/

/\* for(i=1; i!=10; i=i+2)

printf("%d ",i);

\*/

/\*(xiv)\*/

/\* for(j=10; i=j; j-=2)

printf("%d ",i);

\*/

/\*(xv)\*/

/\* for(i=10; i--; )

printf("%d ",i);

\*/

return 0;

}

23.

#include<stdio.h>

int main(void)

{

int term,i;

/\*(i)\*/

for(term=10; term<=70; term+=10)

printf("%d ",term);

printf("\n");

/\*(ii)\*/

for(term=-10; term>=-70; term-=10)

printf("%d ",term);

printf("\n");

/\*(iii)\*/

for(term=1; term<=11111; term=term\*10+1)

printf("%d ",term);

printf("\n");

/\*(iv)\*/

i=0;

for(term=1; term<=16; i++,term=term+i)

printf("%d ",term);

/\*(v)\*/

printf("\n");

for(term=10; term>=-5; term-=3)

printf("%d ",term);

printf("\n");

/\*(vi)\*/

for(term=1; term<=32; term\*=2)

printf("%d ",term);

printf("\n");

return 0;

}

24.

#include<stdio.h>

int main(void)

{

int i;

for(i=1; i<5; i++);

printf("%d ",i);

/\*

i=10;

while(i<5);

printf("%d ",i++);

\*/

/\*

i=0;

while(i<5);

printf("%d",i++);

\*/

return 0;

}

25.

#include<stdio.h>

int main(void)

{

int i=1;

while(i++ < 5)

printf("%d ",i);

printf("\n");

i=1;

while(++i < 5)

printf("%d ",i);

printf("\n");

i=6;

while(i--)

printf("%d ",i);

return 0;

}

26.

#include<stdio.h>

int main(void)

{

int a=6,b=4;

while(a+b)

{

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

a/=2;

b%=3;

}

return 0;

}

27.

#include<stdio.h>

int main(void)

{

int i=10;

do

{

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

i=i-3;

}while(i);

return 0;

}

28.

#include<stdio.h>

int main(void)

{

int i,j,x=0;

for(i=0; i<5; i++)

for(j=i; j>0; j--)

x=i+j+1;

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

return 0;

}

29.

#include<stdio.h>

int main(void)

{

int i,sum;

for(i=0; i<10; i+=3)

sum+=i\*i;

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

return 0;

}

30.

#include<stdio.h>

int main(void)

{

int i=1;

for( ; ; )

if(i++==3)

break;

printf("%d ",i);

i=1;

while( )

if(i++==3)

break;

printf("%d ",i);

return 0;

}

31.

#include<stdio.h>

int main(void)

{

int i,sum1=0,sum2=0;

for(i=1; i<5; i++)

sum1+=i;

i=1;

while(i<5)

{

i++;

sum2+=i;

}

printf("%d %d\n",sum1,sum2);

return 0;

}

32.

#include<stdio.h>

int main(void)

{

int i,j;

for(j=i+1,i=1; i<=5; j++,i++)

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

return 0;

}

33.

#include<stdio.h>

int main(void)

{

int i=1;

do;

while(++i<10);

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

return 0;

}

34.

#include<stdio.h>

int main(void)

{

int i,j,k;

printf("(i)\n");

for(i=1; i<=6; i++)

printf("\*");

printf("\n\n(ii)\n");

for(j=1; j<=5; j++)

for(i=1; i<=6; i++)

printf("\*");

printf("\n");

printf("\n\n(iii)\n");

for(j=1; j<=5; j++)

{

for(i=1;i<=6;i++)

printf("\*");

printf("\n");

}

printf("\n\n(iv)\n");

for(k=1; k<=3; k++)

{

for(j=1; j<=5; j++)

{

for(i=1; i<=6; i++)

printf("\*");

printf("\n");

}

printf("\n");

}

return 0;

}

35.

#include<stdio.h>

#include<math.h>

int main(void)

{

int i,j,n;

for(i=1; ;i++)

{

if(i>5)

break;

printf("%d ",i);

}

printf("\n\n");

i=1,j=3;

while(i<5)

{

if(j==0)

break;

printf("%d ",i++,j--);

}

printf("\n\n");

n=43;

for(i=2; i<=sqrt(n); i++)

if(n%i == 0)

break;

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

return 0;

}

36.

#include<stdio.h>

int main(void)

{

int i=1,j=3;

while(i<5)

{

i++;

if(j==0)

break;

j--;

}

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

return 0;

}

37.

#include<stdio.h>

int main(void)

{

int i,sum,n;

i=1;

for( ; ; )

{

if(i==5)

break;

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

i++;

}

sum=0;

for( ; ; )

{

if(sum>100)

break;

scanf("%d",&n);

sum+=n;

}

return 0;

}

38.

#include<stdio.h>

int main(void)

{

int i,j,k;

i=0,j=10,k=6;

while(1)

{

if(i==j || k<0)

break;

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

i++;

j--;

k=k-2;

}

printf("\n");

for(i=0,j=10,k=6; i!=j && k>=0; i++,j--,k=k-2)

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

return 0;

}

39.

#include<stdio.h>

int main(void)

{

int i,z,emp\_id;

i=1,z=0;

do

{

if(i<5)

{

printf("%d ",i++);

z=z+2;

}

}while(i<5);

printf("\n");

do

{

printf("Enter employee ID(100-500) : ");

scanf("%d", &emp\_id);

}while(emp\_id < 100 || emp\_id > 500);

return 0;

}

40.

#include<stdio.h>

int main(void)

{

int l,b,h,sarea,vol;

printf("Enter length, width,height of a cubiod : ");

scanf("%d%d%d",&l,&b,&h);

while(l<=0 || b<=0 || h<=0)

{

printf("Enter length,width,height of a cubiod : ");

scanf("%d%d%d",&l,&b,&h);

}

sarea=2\*(l\*b + b\*h + h\*l);

vol=l\*b\*h;

printf("Surface Area=%d, Volume=%d\n",sarea,vol);

return 0;

}

41.

#include<stdio.h>

int main(void)

{

int i;

for(i=1; i<=5; i++)

printf("%d ",i);

printf("\n");

i=1;

while(i<=5)

{

printf("%d ",i);

i++;

}

printf("\n");

i=1;

do

{

printf("%d ",i);

i++;

}while(i<=5);

printf("\n");

return 0;

}

42.

#include<stdio.h>

int main(void)

{

int i;

for(i=1; i<=5; i++)

{

if(i==3)

continue;

printf("%d ",i);

}

printf("\n");

i=1;

while(i<=5)

{

if(i==3)

continue;

printf("%d ",i);

i++;

}

printf("\n");

i=1;

do

{

if(i==3)

continue;

printf("%d ",i);

i++;

}while(i<=5);

return 0;

}

43.

#include<stdio.h>

int main(void)

{

int i=0;

while(i<=5)

i++;

printf("%d ",i);

return 0;

}

44.

#include<stdio.h>

int main(void)

{

int i,x,sum;

for(i=1; i<6; i++)

{

if(i==3)

continue;

printf("%d ",i);

}

printf("\n");

x=0,sum=0;

while(x<10)

{

x++;

if(x%2==0)

continue;

sum+=x;

}

printf("%d ",sum);

printf("\n");

return 0;

}

45.

#include<stdio.h>

/\*(i)\*/

int main(void)

{

int i,j,k;

for(i=1; i<8; i++)

{

if(i!=3 && i!=5)

{

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

{

for(k=j; k>0; k--)

printf("%d ",k);

printf("\n");

}

printf("\n");

}

}

return 0;

}

46.

#include<stdio.h>

int main(void)

{

int i;

for(i=1; i<=80; i++)

{

if(i%8==0)

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

else

printf("%2d\t",i);

}

return 0;

}

47.

#include<stdio.h>

int main(void)

{

int i,n,large;

printf("Input 10 numbers : \n");

scanf("%d",&n);

large=n;

for(i=2; i<=10; i++)

{

scanf("%d",&n);

if(n > large)

large=n;

}

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

return 0;

}

48.

#include<stdio.h>

int main(void)

{

int i,j,n;

for(i=1; i<=10; i++)

{

scanf("%d",&n);

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

printf("=");

printf("\n");

}

return 0;

}

49.

#include<stdio.h>

#include<math.h>

int main(void)

{

int i,num;

for(num=2; num<=99; num++)

{

for(i=2; i<=sqrt(num); i++)

if(num%i == 0)

break;

if(i>sqrt(num))

printf("%d\t",num);

}

return 0;

}

50.

#include<stdio.h>

int main(void)

{

int n,d,rev;

printf("Enter a number : ");

scanf("%d",&n);

rev=0;

do

{

d=n%10;

n/=10;

rev=rev\*10+d;

}while(n>0);

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

return 0;

}

51.

#include<stdio.h>

int main(void)

{

int n,d,rev;

printf("Enter a number : ");

scanf("%d",&n);

rev=0;

do

{

d=n%10;

n/=10;

rev=rev\*10+d;

}while(n>0);

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

return 0;

}

52.

#include<stdio.h>

#include<math.h>

int main(void)

{

int num,n,d,rev;

for(n=1000; n<=9999; n++)

{

num=n; rev=0;

do

{

d=num%10;

num/=10;

rev=rev\*10+d;

}while(num>0);

if(n==rev)

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

}

return 0;

}

53.

#include<stdio.h>

int main(void)

{

int n,count=0,rem;

printf("Enter a number : ");

scanf("%d", &n);

n/=10;

count++;

while(n>0)

{

n/=10;

count++;

}

printf("Number of digits=%d\n", count);

return 0;

}

54.

#include<stdio.h>

int main(void)

{

int n,count=0,dig,d;

printf("Enter a number : ");

scanf("%d",&n);

printf("Enter a digit : ");

scanf("%d",&dig);

do

{

d=n%10;

n/=10;

if(d==dig)

count++;

}while(n>0);

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

return 0;

}

55.

#include<stdio.h>

int main(void)

{

int i,n,sum;

printf("Enter n : ");

scanf("%d",&n);

sum=0;

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

sum+=i\*i;

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

return 0;

}

56.

#include<stdio.h>

int main(void)

{

int i,n,sum;

printf("Enter n : ");

scanf("%d",&n);

sum=0;

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

if(i%3==0)

sum+=i\*i\*i;

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

/\*using continue\*/

sum=0;

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

{

if(i%3!=0)

continue;

sum+=i\*i\*i;

}

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

return 0;

}

57.

#include<stdio.h>

int main(void)

{

int x,y,i,result;

printf("Enter x and y : ");

scanf("%d%d",&x,&y);

result=1;

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

result\*=x;

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

return 0;

}

58.

#include<stdio.h>

int main(void)

{

long x,y,z,num;

printf("Enter a number : ");

scanf("%d",&num);

x=0; y=1;

do

{

z = x+y;

x = y;

y = z;

}while(z<num);

if(z==num)

printf("%d is a Fibonacci number\n",num);

else

printf("%d is not a Fibonacci number\n",num);

return 0;

}

59.

#include<stdio.h>

int main(void)

{

int i,j,n,x,term,sum;

printf("Enter n : ");

scanf("%d",&n);

sum=0;term=1;

for(i=1; i<=n; i++,term=term\*10+1)

{

printf("%d+",term);

sum+=term;

}

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

printf("Enter x : ");

scanf("%d",&x);

sum=0;

for(i=1,term=x; i<=n; i++,term\*=x)

{

printf("%d + ",term);

sum+=term;

}

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

sum=0;

for(i=1,term=x; i<=n; i++,term\*=-x)

{

printf("%d + ",term);

sum+=term;

}

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

return 0;

}

60.

#include<stdio.h>

int main(void)

{

int i,m1,m2,m3,total,rno,maxtotal=0,maxrno,count;

count=0;

for(i=1; i<=10; i++)

{

do

{

printf("Enter roll number(1000-9999) : ");

scanf("%d",&rno);

}while(rno < 1000 || rno > 9999 );

printf("Enter marks of three subjects : ");

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

if( m1<40 || m2<40 || m3<40 )

continue;

total=m1+m2+m3;

if(total > 200)

count++;

if(maxtotal==0 || total > maxtotal)/\*maxtotal==0 should be written first\*/

{

maxtotal=total;

maxrno=rno;

}

}

printf("%d students got more than 200 marks\n",count);

printf("Student with roll no. %d got highest marks %d\n",maxrno,maxtotal);

return 0;

}

61.

#include<stdio.h>

int main(void)

{

int i,j,n;

printf("Enter n : ");

scanf("%d",&n);

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

{

if(i%2==0)

printf(" ");

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

printf("\* ");

printf("\n");

}

return 0;

}

62.

#include<stdio.h>

main()

{

int i,j,n;

printf("Enter n :");

scanf("%d",&n);

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

{

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

printf("\*");

printf("\n");

}

printf("\n\n");

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

{

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

if(i==1 || i==n || j==1 || j==n)

printf("\*");

else

printf(" ");

printf("\n");

}

printf("\n\n");

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

{

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

{

if(i==j)

printf("\\");

else if(i+j==n+1)

printf("/");

else

printf("\*");

}

printf("\n");

}

printf("\n\n");

return 0;

}

Exercise-Functions

1.

#include<stdio.h>

void func(void);

int main(void)

{

printf("Lucknow\n");

goto ab;

return 0;

}

void func(void)

{

ab:

printf("Bareilly\n");

}

2.

#include<stdio.h>

void func(int a,int b);

int main(void)

{

int x;

x=func(2,3);

return 0;

}

void func(int a,int b)

{

int s;

s=a+b;

return;

}

3.

#include<stdio.h>

int func(void);

int main(void)

{

int x=10;

x=func();

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

return 0;

}

int func(void)

{

printf("Function\n");

}

4.

#include<stdio.h>

int add(int x,y,z)

{

return x+y+z;

}

int main(void)

{

int sum;

sum=add(1,2,3);

return 0;

}

5.

#include<stdio.h>

int main(void)

{

int s;

s=func(1,2);

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

s=func(1,2,3,4);

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

return 0;

}

int func(int a,int b,int c)

{

return a+b+c;

}

6.

#include<stdio.h>

int sum(int x,int y);

int multiply(int x,int y);

int main(void)

{

int m=6,n=3;

printf("%d\t",multiply(m,n));

printf("%d\t",multiply(15,4));

printf("%d\t",multiply(m+n,m-n));

printf("%d\n",multiply(6,sum(m,n)));

return 0;

}

int multiply(int x,int y)

{

return x\*y;

}

int sum(int x,int y)

{

return x+y;

}

7.

#include<stdio.h>

int func(int x,int y);

int main(void)

{

int p=func(5,6);

printf("%d",p);

return 0;

}

int func(int x,int y)

{

int x=2;

return x\*y;

}

8.

#include<stdio.h>

int min(int a,int b);

int main(void)

{

int a=10,b=5;

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

return 0;

}

int min(int a,int b)

{

a<b? return a: return b;

}

9.

#include<stdio.h>

int max(int a,int b)

{

return a>b? a: b;

}

int main(void)

{

int a=2,b=8,c=3;

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

return 0;

}

10.

#include<stdio.h>

void func(int x,int y);

int main(void)

{

int x;

x=func(5,6)+100;

printf("%d",x);

return 0;

}

void func(int x,int y)

{

int z;

z=x+y;

}

11.

#include<stdio.h>

int diff(int x,int y)

{ return x-y; }

int sum(int x,int y)

{ return x+y; }

int main(void)

{

int a=20,b=5,c=2,d=6;

printf("%d\t",a+diff(d,c));

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

return 0;

}

12.

#include<stdio.h>

int sqr(int x);

int cube(int x);

int func(int n);

int main(void)

{

int n=5;

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

return 0;

}

int sqr(int x)

{ return x\*x; }

int cube(int x)

{ return x\*x\*x; }

int func(int n)

{

return n+sqr(n-2)+cube(n-3);

}

13.

#include<stdio.h>

int func(int a,int b,int c);

int main(void)

{

int x=1,y=2,z=3,result;

result = func(x, y, (z=5,z+10));

printf("x=%d, y=%d, z=%d",x,y,z);

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

return 0;

}

func(int a, int b, int c)

{

return 2\*(a+b+c);

}

14.

#include<stdio.h>

int sum(int a,int b);

int main(void)

{

(void)sum(1,2);

return 0;

}

int sum(int a,int b)

{

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

return a+b;

}

15.

#include<stdio.h>

int square1(int a);

int square2(double a);

double square3(int a);

double square4(double a);

int main(void)

{

double x=2.5,y;

y=square1(x);

printf("%lf\t",y);

y=square2(x);

printf("%lf\t",y);

y=square3(x);

printf("%lf\t",y);

y=square4(x);

printf("%lf\n",y);

return 0;

}

int square1(int a)

{

return a\*a;

}

int square2(double a)

{

return a\*a;

}

double square3(int a)

{

return a\*a;

}

double square4(double a)

{

return a\*a;

}

16.

#include<stdio.h>

main()

{

int func(int a,int b)

{

return (a+b);

}

int c;

c=func(3,5);

printf("%d",c);

return 0;

}

17.

#include<stdio.h>

void display(int,int);

int main(void)

{

int x=15;

float y=290.5;

display(x,y);

return 0;

}

void display(int a,int b)

{

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

}

18.

#include<stdio.h>

void func(void);

int main(void)

{

int i=5;

for(i=i+1; i<8; i++)

func();

return 0;

}

void func(void)

{

int j;

for(j=1; j<3; j++)

printf("%d\t",++j);

}

19.

#include<stdio.h>

int func(int a,int b);

int main(void)

{

int i=2,j=3;

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

return 0;

}

int func(int a,int b)

{

a=a-5;

b=b+5;

return(!a + --b);

}

20.

#include<stdio.h>

int func(int a,int b,int c);

int main(void)

{

int x;

x=func(2,3,4);

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

return 0;

}

int func(int a,int b,int c)

{

return a,b,c;

}

21.

#include<stdio.h>

void func(int a,int b);

int main(void)

{

int i=5,j=10;

func(i/2,j%3);

return 0;

}

void func(int a,int b)

{

a/=2;

b--;

printf("%d\t",a+b);

}

22.

#include<stdio.h>

int a=5;

void func(void);

int main(void)

{

func();

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

return 0;

}

void func(void)

{

int a=2;

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

}

23.

#include<stdio.h>

int func(int x,int y);

int main(void)

{

int a=2,b=5;

a=func(a+b,a-b);

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

return 0;

}

int func(int x,int y)

{

return x+y,x-y;

}

24.

#include<stdio.h>

int func(int k);

int main(void)

{

int i=0,k=3;

i+=func(k);

i+=func(k);

i+=func(k);

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

return 0;

}

int func(int k)

{

static int m=2;

m=m+k;

return m;

}

25.

#include<stdio.h>

int main(void)

{

int i=9;

if(i==9)

{

int i=25;

}

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

return 0;

}

26.

#include<stdio.h>

void func(int a,static int b);

int main(void)

{

func(1,2);

func(3,4);

return 0;

}

void func(int a,static int b)

{

a++;

b++;

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

}

27.

#include<stdio.h>

int func(void);

int main(void)

{

int i;

for(i=1; i<=6; i++)

printf("%d ",func());

return 0;

}

int func(void)

{

static int k=1;

k\*=2;

return k;

}

28.

#include<stdio.h>

int func(int n);

int main(void)

{

printf("%d ",func(2));

printf("%d ",func(5));

printf("%d ",func(2));

return 0;

}

int func(int n)

{

static int s=0;

int i;

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

s+=i;

return s;

}

29.

#include<stdio.h>

int func(int a,int b);

int func1(int a, int b);

int func2(int a,int b);

int f1(int x,int y);

int f2(int x,int y);

int main(void)

{

int a,b;

printf("Enter a and b : ");

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

printf("%d ",func(a,b));

printf("%d ",func1(a,b));

printf("%d ",func2(a,b));

return 0;

}

int func(int a,int b)

{

int x;

if(a==b)

x=f1(a,b);

else

x=f2(a,b);

return x;

}

int func1(int a,int b)

{

if(a==b)

return f1(a,b);

return f2(a,b);

}

int func2(int a,int b)

{

return a==b? f1(a,b): f2(a,b);

}

int f1(int x,int y)

{

return x+y;

}

int f2(int x,int y)

{

return x-y;

}

30.

#include<stdio.h>

#include<math.h>

int func(int n);

int func1(int n);

int main(void)

{

int n;

printf("Enter a number : ");

scanf("%d",&n);

printf("%d ",func(n));

printf("%d ",func1(n));

return 0;

}

int func(int n)

{

int i,flag=0;

for(i=2; i<=sqrt(n); i++)

{

if(n%i==0)

{

flag=1;

break;

}

}

return flag;

}

int func1(int n)

{

int i;

for(i=2; i<=sqrt(n); i++)

if(n%i==0)

return 1;

return 0;

}

31.

#include<stdio.h>

int func(int n);

int func1(int n);

int main(void)

{

int n;

printf("Enter n : ");

scanf("%d",&n);

printf("%d ",func(n));

printf("%d ",func1(n));

return 0;

}

int func(int n)

{

if(n==39)

{

n+=5;

return n;

}

else

return n;

}

int func1(int n)

{

if(n==39)

n+=5;

return n;

}

32.

#include<stdio.h>

int func(int x,int b);

int func1(int x,int b);

int main(void)

{

int x,b;

printf("Enter x and b : ");

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

printf("%d ",func(x,b));

printf("%d ",func1(x,b));

return 0;

}

int func(int x,int b)

{

if(x==5)

return b+2;

else if(x>5)

return b+10;

else

return b;

}

int func1(int x,int b)

{

if(x==5)

b+=2;

else if(x>5)

b+=10;

return b;

}

33.

#include<stdio.h>

int abs(int a);

int abs1(int a);

int main(void)

{

int a;

printf("Enter a : ");

scanf("%d",&a);

printf("%d ",abs(a));

printf("%d ",abs1(a));

return 0;

}

int abs(int a)

{

if(a<0)

return -a;

else

return a;

}

int abs1(int a)

{

if(a<0)

a=-a;

return a;

}

34.

#include<stdio.h>

int func(int a,int b);

int func1(int a,int b);

int func2(int a,int b);

int main(void)

{

int a,b;

printf("Enter a and b : ");

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

printf("%d ",func(a,b));

printf("%d ",func1(a,b));

printf("%d ",func2(a,b));

return 0;

}

int func(int a,int b)

{

a=a-b;

if(a!=0)

return a;

else

return 0;

}

int func1(int a,int b)

{

a=a-b;

return a;

}

int func2(int a,int b)

{

return a-b;

}

35.

#include<stdio.h>

int func(int,int);

int func1(int,int);

int main(void)

{

int a,b;

printf("Enter a and b :");

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

printf("%d ",func(a,b));

printf("%d ",func1(a,b));

return 0;

}

int func(int a,int b)

{

a=a-b;

if(a!=0)

{

a++;

return a;

}

else

return 0;

}

int func1(int a,int b)

{

a=a-b;

if(a!=0)

a++;

return a;

}

36.

#include<stdio.h>

int reverse(int n);

int isPalindrome(int num);

int isPalindrome1(int num);

int main(void)

{

int num;

printf("Enter a number : ");

scanf("%d",&num);

if( isPalindrome(num) )

printf("Number is a palindrome\n");

else

printf("Number is not a palindrome\n");

if( isPalindrome1(num) )

printf("Number is a palindrome\n");

else

printf("Number is not a palindrome\n");

return 0;

}

int isPalindrome(int num)

{

if(num==reverse(num))

return 1;

else

return 0;

}

int isPalindrome1(int num)

{

if(num==reverse(num))

return 1;

return 0;

}

int reverse(int n)

{

int rev=0;

do

{

rev=rev\*10+n%10;

n/=10;

}while(n>0);

return rev;

}

37.

#include<stdio.h>

int func(int a,int b);

int func1(int a,int b);

int main(void)

{

int a,b;

printf("Enter a and b : ");

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

printf("%d %d ",func(a,b),func1(a,b));

return 0;

}

int func(int a,int b)

{

a=a-b;

if(a!=2)

{

a++;

return a;

}

else

{

a=a+b;

printf("%d ",a);

return b;

}

}

int func1(int a,int b)

{

a=a-b;

if(a!=2)

{

a++;

return a;

}

a=a+b;

printf("%d ",a);

return b;

}

38.

#include<stdio.h>

int func(int,int);

int func1(int,int);

int main(void)

{

int a,b;

printf("Enter a and b : ");

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

printf("%d %d ",func(a,b),func1(a,b));

return 0;

}

int func(int m,int n)

{

if(m==5)

return n-10;

else if(m>5)

return n+10;

else

return n;

}

int func1(int m,int n)

{

if(m==5)

return n-10;

if (m>5)

return n+10;

return n;

}

39.

#include<stdio.h>

char findGrade(int m1,int m2,int m3,int m4);

char findGrade1(int m1,int m2,int m3,int m4);

int main(void)

{

int m1,m2,m3,m4;

printf("Enter m1,m2,m3,m4 : ");

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

printf("%c ",findGrade(m1,m2,m3,m4));

printf("%c ",findGrade1(m1,m2,m3,m4));

return 0;

}

char findGrade(int m1,int m2,int m3,int m4)

{

float total,per;

total = m1+m2+m3+m4;

per = total/4;

if(per>=85)

return 'A';

else if(per>=70)

return 'B';

else if(per>=55)

return 'C';

else if(per>=40)

return 'D';

else

return 'E';

}

char findGrade1(int m1,int m2,int m3,int m4)

{

float total,per;

total = m1+m2+m3+m4;

per = total/4;

if(per>=85)

return 'A';

if(per>=70)

return 'B';

if(per>=55)

return 'C';

if(per>=40)

return 'D';

return 'E';

}

40.

#include<stdio.h>

void mult\_table(num);

int main(void)

{

int num;

printf("Enter a number : ");

scanf("%d",&num);

mult\_table(num);

return 0;

}

void mult\_table(num)

{

int i;

for(i=1; i<=10; i++)

printf("%2d x %2d = %3d\n", num, i, num\*i);

}

41.

#include<stdio.h>

void PrintArmstrong(void);

int isArmstrong(int num);

int cubesum(int n);

int main(void)

{

int num;

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

PrintArmstrong();

printf("Enter a number : ");

scanf("%d",&num);

if(isArmstrong(num))

printf("%d is an Armstrong number\n",num);

else

printf("%d is not an Armstrong number\n",num);

return 0;

}

void PrintArmstrong(void)

{

int num;

for(num=100; num<=999; num++)

if(num==cubesum(num))

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

}

int isArmstrong(int num)

{

if(num==cubesum(num))

return 1;

else

return 0;

}

int cubesum(int n)

{

int d,cube,sum=0;

while(n>0)

{

d = n%10;

n/=10;

cube = d\*d\*d;

sum = sum+cube;

}

return sum;

}

42.

#include<stdio.h>

int ProdDigits(long num);

int main(void)

{

long num;

printf("Enter a number : ");

scanf("%ld",&num);

printf("%d\n",ProdDigits(num));

return 0;

}

int ProdDigits(long num)

{

int prod=1,dig;

do

{

dig=num%10;

prod\*=dig;

num/=10;

}while(num>0);

return prod;

}

43.

#include<stdio.h>

int MDR(long int num);

int mPersistence(long int num);

int ProdDigits(long int num);

int main(void)

{

long int num;

printf("Enter a number : ");

scanf("%ld",&num);

printf("Multiplicative Digital root of %d is %d\n",num,MDR(num));

printf("Persistence of %d is %d\n",num,mPersistence(num));

return 0;

}

int MDR(long int num)

{

while(num>9)/\*while num is more than one digit\*/

num=ProdDigits(num);

return num;

}

int mPersistence(long int num)

{

int per=0;

while(num>9)/\*while num is more than one digit\*/

{

per++;

num=ProdDigits(num);

}

return per;

}

int ProdDigits(long num)

{

int prod=1,dig;

do

{

dig=num%10;

prod\*=dig;

num/=10;

}while(num>0);

return prod;

}

44.

#include<stdio.h>

int sumPrDivisors(int i);

int main(void)

{

int i;

printf("Enter i : ");

scanf("%d",&i);

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

return 0;

}

int sumPrDivisors(int num)

{

int i,s=0;

for(i=1; i<=num/2; i++)

if(num%i==0)

s+=i;

return s;

}

45.

#include<stdio.h>

int sumPrDivisors(int i);

int main(void)

{

int i;

for(i=1; i<=10000; i++)

if( sumPrDivisors(i)==i )

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

return 0;

}

int sumPrDivisors(int num)

{

int i,s=0;

for(i=1;i<=num/2;i++)

if(num%i==0)

s+=i;

return s;

}

46.

#include<stdio.h>

int sumPrDivisors(int i);

int main(void)

{

int i,j;

for(i=1; i<=2000; i++)

for(j=1; j<=2000; j++)

if( i!=j && sumPrDivisors(i)==j && sumPrDivisors(j)==i )

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

return 0;

}

int sumPrDivisors(int num)

{

int i,s=0;

for(i=1; i<=num/2; i++)

if(num%i==0)

s+=i;

return s;

}

46.

#include<stdio.h>

int sumPrDivisors(int i);

void perfect\_amicable(int x,int y);

int main(void)

{

perfect\_amicable(1,2000);

return 0;

}

void perfect\_amicable(int x,int y)

{

int i,j;

for(i=x; i<=y; i++)

for(j=x; j<=y; j++)

if( sumPrDivisors(i)==j && sumPrDivisors(j)==i )

{

if(i==j)

printf("Perfect number : %d\n", i);

else

printf("Amicable numbers : %d %d\n",i,j);

}

}

int sumPrDivisors(int num)

{

int i,s=0;

for(i=1;i<=num/2;i++)

if(num%i==0)

s+=i;

return s;

}

48.

#include<stdio.h>

int isPrime(int n);

int sumPrDivisors(int i);

int main(void)

{

int i;

for(i=1; i<=100; i++)

if( isPrime(i))

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

return 0;

}

int isPrime(int n)

{

return sumPrDivisors(n)==1;

}

int sumPrDivisors(int num)

{

int i,s=0;

for(i=1; i<=num/2; i++)

if(num%i==0)

s+=i;

return s;

}

49.

#include<stdio.h>

float convert(float n, char unitType);

int main(void)

{

float len;

printf("Enter length in inches : ");

scanf("%f",&len);

printf("Length in cms = %f\n", convert(len, 'i'));

printf("Enter length in cms : ");

scanf("%f",&len);

printf("Length in inches = %f\n", convert(len, 'c') );

return 0;

}

float convert(float n, char unitType)

{

return unitType=='i' ? n\*2.54 : n/2.54;

}

50.

#include<stdio.h>

int Product(int a,int b);

int Quotient(int a,int b);

int Remainder(int a,int b);

int main(void)

{

int x,y;

printf("Enter x and y (y should be non-zero) : ");

scanf("%d%d",&x,&y);

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

printf("Quotient=%d\n",Quotient(x,y));

printf("Remainder=%d\n",Remainder(x,y));

return 0;

}

int Product(int a, int b)

{

int i,p=0;

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

p+=a;

return p;

}

int Quotient(int a, int b)

{

int q=0;

while(a>=b)

{

a-=b;

q++;

}

return q;

}

int Remainder(int a, int b)

{

while(a>=b)

a-=b;

return a;

}

51.

#include<stdio.h>

int isAlphaNumeric(char ch);

int isAlphabet(char ch);

int isNumeric(char ch);

int main(void)

{

char ch;

printf("Enter a character : ");

scanf("%c",&ch);

if( isAlphaNumeric(ch) )

printf("%c is AlphaNumeric\n",ch);

else

printf("%c is not AlphaNumeric\n",ch);

return 0;

}

int isAlphaNumeric(char ch)

{

return isAlphabet(ch) || isNumeric(ch);

}

int isAlphabet(char ch)

{

return ch>=65 && ch<=90 || ch>=97 && ch<=122;

}

int isNumeric(char ch)

{

return ch>=48 && ch<=57;

}

52.

#include<stdio.h>

char upper(char ch);

int main(void)

{

char ch;

while((ch=getchar())!='\n')

putchar(upper(ch));

return 0;

}

char upper(char ch)

{

return (ch>=97 && ch<=122) ? ch-32 : ch;

}

53.

#include<stdio.h>

double series(int n);

int main(void)

{

int n;

printf("Enter number of terms : ");

scanf("%d",&n);

printf("%lf\n",series(n));

return 0;

}

double series(int n)

{

int i;

double term,sum=0;

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

{

term=1.0/(i\*i);/\*should take 1.0 for floating point arithmetic\*/

sum+=term;

}

return sum;

}

54.

#include<stdio.h>

#include<math.h>

int isprime(int n);

void PrintTwinPrimes(int x);

void PrintTwinPrimes1(int x);

int main(void)

{

PrintTwinPrimes(1000);

PrintTwinPrimes1(1000);

return 0;

}

void PrintTwinPrimes(int n)

{

int i;

for(i=3; i<n; i=i+2)

{

if(isprime(i) && isprime(i+2) )

printf("%5d %5d\n",i,i+2);

}

}

/\*isprime() callled lesser times in this function\*/

void PrintTwinPrimes1(int n)

{

int i,flag=1;

for(i=3; i<n; i=i+2)

{

if(isprime(i+2))

{

if(flag==1)

printf("%5d %5d\n",i,i+2);

else

flag=1;

}

else

flag=0;

}

}

int isprime(int n)

{

int i;

for(i=2; i<=sqrt(n); i++)

if(n%i==0)

return 0;

return 1;

}

54.

#include<stdio.h>

int isLeap(int y);

int main(void)

{

int year;

printf("Enter year : ");

scanf("%d",&year);

if(isLeap(year))

printf("%d is a Leap Year\n", year);

else

printf("%d is not a Leap Year\n", year);

return 0;

}

int isLeap(int y)

{

if(y%100!=0 && y%4==0 || y%400==0)

return 1;

return 0;

}

56.

#include<stdio.h>

int isValid(int d,int m,int y);

int isLeap(int y);

int main(void)

{

int day,month,year;

printf("Enter date(dd/mm/yyyy): ");

scanf("%d/%d/%d",&day,&month,&year);

if( isValid(day,month,year) )

printf("%d/%d/%d is a valid date\n",day,month,year);

else

printf("%d/%d/%d is not a valid date\n",day,month,year);

return 0;

}

int isValid(int d,int m,int y)

{

if(y<1850 || y>2050 || m<1 || m>12 || d<1 || d>31)

return 0;

if(m==2) /\*Check for number of days in February\*/

if(d==30 || d==31 || (d==29 && !isLeap(y)))

return 0;

if(m==4 || m==6 || m==9 || m==11)/\*Check days in April,June,Sept,Nov\*/

if(d==31)

return 0;

return 1;

}

int isLeap(int y)

{

if(y%100!=0 && y%4==0 || y%400==0)

return 1;

return 0;

}

57.

#include<stdio.h>

int main(void)

{

int d1,m1,y1,d2,m2,y2,c;

printf("Enter Date1(dd/mm/yyyy): ");

scanf("%d/%d/%d",&d1,&m1,&y1);

printf("Enter Date2(dd/mm/yyyy): ");

scanf("%d/%d/%d",&d2,&m2,&y2);

c=cmpdate(d1,m1,y1,d2,m2,y2);

if(c==1)

printf("Date 1 is before Date 2\n");

else if(c==-1)

printf("Date 2 is before Date 1\n");

else

printf("Dates are same\n");

return 0;

}

int cmpdate(int d1, int m1, int y1, int d2, int m2, int y2)

{

if(y1<y2)

return 1;

if(y1>y2)

return -1;

if(m1<m2)

return 1;

if(m1>m2)

return -1;

if(d1<d2)

return 1;

if(d1>d2)

return -1;

return 0;

}

58.

#include<stdio.h>

int findJulian(int d,int m,int y);

int isLeap(int y);

int main(void)

{

int day,month,year;

printf("Enter date(dd/mm/yyyy): ");

scanf("%d/%d/%d",&day,&month,&year);

printf("Julian Day = %d\n",findJulian(day,month,year));

return 0;

}/\*End of main()\*/

int findJulian(int d,int m,int y)

{

int j=d;

switch(m-1)

{

case 11: j+=30; /\*Fall thru in all cases\*/

case 10: j+=31;

case 9: j+=30;

case 8: j+=31;

case 7: j+=31;

case 6: j+=30;

case 5: j+=31;

case 4: j+=30;

case 3: j+=31;

case 2: j+=28;

case 1: j+=31;

}

if( isLeap(y) && m>2 )

j=j+1;

return j;

}

int isLeap(int y)

{

if(y%100!=0 && y%100!=0 || y%400==0)

return 1;

return 0;

}

59.

#include<stdio.h>

void printDayOfWeek(int day,int month,int year);

int findJulian(int day, int month, int year);

int isLeap(int year);

void Print(int julianDay,int year);

int isValid(int d,int m,int y);

int main(void)

{

int day,month,year;

printf("Enter date(dd/mm/yyyy): ");

scanf("%d/%d/%d",&day,&month,&year);

if(isValid(day,month,year))

printDayOfWeek(day, month,year);

else

printf("Not a Valid date\n");

return 0;

}/\*End of main()\*/

void printDayOfWeek(int day,int month,int year)

{

int julianDay;

julianDay = findJulian(day,month,year);

Print(julianDay,year);

}

void Print(int j,int y)

{

int f,h,fh,day;

f = (y-1)/4;

h = (y-1)/100;

fh = (y-1)/400;

day = (y+j+f-h+fh)%7;

switch(day)

{

case 0: printf("Saturday\n"); break;

case 1: printf("Sunday\n"); break;

case 2: printf("Monday\n"); break;

case 3: printf("Tuesday\n"); break;

case 4: printf("Wednesday\n"); break;

case 5: printf("Thursday\n"); break;

case 6: printf("Friday\n"); break;

}

}

int findJulian(int d,int m,int y)

{

int j=d;

switch(m-1)

{

case 11: j+=30; /\*Fall thru in all cases\*/

case 10: j+=31;

case 9: j+=30;

case 8: j+=31;

case 7: j+=31;

case 6: j+=30;

case 5: j+=31;

case 4: j+=30;

case 3: j+=31;

case 2: j+=28;

case 1: j+=31;

}

if( isLeap(y) && m>2 )

j=j+1;

return j;

}

int isLeap(int y)

{

if(y%100!=0 && y%4==0 || y%400==0)

return 1;

return 0;

}

int isValid(int d, int m, int y)

{

if(y<1850 || y>2050 || m<1 || m>12 || d<1 || d>31)

return 0;

if(m==2) /\*Check for number of days in February\*/

if(d==30 || d==31 || (d==29 && !isLeap(y)))

return 0;

if(m==4 || m==6 || m==9 || m==11)/\*Check days in April,June,Sept,Nov\*/

if(d==31)

return 0;

return 1;

}

60.

#include<stdio.h>

int diffDays(int d1,int m1,int y1,int d2,int m2,int y2);

int findJulian(int day,int month,int year);

int isLeap(int year);

int cmpdate(int d1,int m1,int y1,int d2,int m2,int y2);

int isValid(int d, int m, int y);

int main(void)

{

int d1,m1,y1,d2,m2,y2;

do

{

printf("Dates between years 1850 and 2050 are valid.\n");

printf("Enter dates with valid day,month and year values\n");

printf("Date1 should be before Date2\n\n");

printf("Enter Date1(dd/mm/yyyy): ");

scanf("%d/%d/%d",&d1,&m1,&y1);

printf("Enter Date2(dd/mm/yyyy): ");

scanf("%d/%d/%d",&d2,&m2,&y2);

}while( !isValid(d1,m1,y1) || !isValid(d2,m2,y2) || cmpdate(d1,m1,y1,d2,m2,y2)==-1 );

printf("Difference = %d\n",diffDays(d1,m1,y1,d2,m2,y2));

return 0;

}/\*End of main()\*/

int cmpdate(int d1, int m1, int y1, int d2, int m2, int y2)

{

if(y1<y2)

return 1;

if(y1>y2)

return -1;

if(m1<m2)

return 1;

if(m1>m2)

return -1;

if(d1<d2)

return 1;

if(d1>d2)

return -1;

return 0;

}

int isValid(int d, int m, int y)

{

if(y<1850 || y>2050 || m<1 || m>12 || d<1 || d>31)

return 0;

if(m==2) /\*Check for number of days in February\*/

if(d==30 || d==31 || (d==29 && !isLeap(y)) )

return 0;

if(m==4 || m==6 || m==9 || m==11)/\*Check days in April,June,Sept,Nov\*/

if(d==31)

return 0;

return 1;

}

int diffDays(int d1,int m1,int y1,int d2,int m2,int y2)

{

int j1,j2,d,y;

j1=findJulian(d1, m1, y1);

j2=findJulian(d2, m2, y2);

if(y1==y2)

return j2-j1;

d=0;

for(y=y1+1; y<=y2-1; y++)

{

if(isLeap(y))

d=d+366;

else

d=d+365;

}

if (isLeap(y1))

return (366-j1) + d + j2 ;

else

return (365-j1) + d + j2 ;

}/\*End of diffDays()\*/

int findJulian(int d, int m, int y)

{

int j = d;

switch(m-1)

{

case 11: j+=30; /\*Fall thru in all cases\*/

case 10: j+=31;

case 9: j+=30;

case 8: j+=31;

case 7: j+=31;

case 6: j+=30;

case 5: j+=31;

case 4: j+=30;

case 3: j+=31;

case 2: j+=28;

case 1: j+=31;

}

if( isLeap(y) && m!=1 && m!=2 )

j = j+1;

return j;

}

int isLeap(int y)

{

if(y%100!=0 && y%4==0 || y%400==0)

return 1;

else

return 0;

}

Exercise- Recursion

1.

#include<stdio.h>

int func1(int a,int b);

int func2(int a,int b);

int main(void)

{

printf("%d %d\n",func1(3,8),func2(3,8));

return 0;

}

func1(int a,int b)

{

if(a>b)

return 0;

return b + func1(a,b-1);

}

func2(int a,int b)

{

if(a>b)

return 0;

return a + func2(a+1,b);

}

2.

#include<stdio.h>

int func(int a, int b);

int main(void)

{

printf("%d \n",func(3,8));

return 0;

}

int func(int a, int b)

{

if(a>b)

return 1000;

return a + func(a+1,b);

}

3.

#include<stdio.h>

int func(int a);

int func1(int a);

int main(void)

{

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

printf("%d\n",func1(6));

return 0;

}

int func(int a)

{

if(a==10)

return a;

return a + func(a+1);

}

int func1(int a)

{

if(a==0)

return a;

return a + func1(a+1);

}

4.

#include<stdio.h>

int func(int a,int b);

int main(void)

{

printf("%d\n",func(4,8));

printf("%d\n",func(3,8));

return 0;

}

int func(int a,int b)

{

if(a==b)

return a;

return a+b+func(a+1,b-1);

}

5.

#include<stdio.h>

void func1(int,int);

void func2(int,int);

int main(void)

{

func1(10,18);

printf("\n");

func2(10,18);

return 0;

}/\*End of main()\*/

void func1(int a,int b)

{

if(a>b)

return;

printf("%d ",b);

func1(a,b-1);

}

void func2(int a,int b)

{

if(a>b)

return;

func2(a,b-1);

printf("%d ",b);

}

6.

#include<stdio.h>

void func1(int a,int b);

void func2(int a,int b);

int main(void)

{

func1(10,18);

printf("\n");

func2(10,18);

return 0;

}

void func1(int a,int b)

{

if(a>b)

return;

printf("%d ",a);

func1(a+1,b);

}

void func2(int a,int b)

{

if(a>b)

return;

func2(a+1,b);

printf("%d ",a);

}

7.

#include<stdio.h>

int func(int a, int b);

int main(void)

{

printf("%d\t",func(3,8));

printf("%d\t",func(3,0));

printf("%d\n",func(0,3));

return 0;

}

int func(int a, int b)

{

if(b==0)

return 0;

if(b==1)

return a;

return a + func(a,b-1);

}

8.

#include<stdio.h>

int count(int n);

int main(void)

{

printf("%d\n",count(17243));

return 0;

}

int count(int n)

{

if(n==0)

return 0;

return 1 + count(n/10);

}

9.

#include<stdio.h>

int func(int n);

int main(void)

{

printf("%d\n",func(14837));

return 0;

}

int func(int n)

{

return (n)? n%10 + func(n/10) : 0;

}

10.

#include<stdio.h>

int count(long int n, int d);

int main(void)

{

printf("%d\n",count(123212,2));

return 0;

}

int count(long int n, int d)

{

if(n == 0)

return 0;

else if(n%10 == d)

return 1 + count(n/10,d);

else

return count(n/10,d);

}

11.

#include<stdio.h>

int f(char \*s,char a);

int main(void)

{

char str[100],a;

printf("Enter a string :");

gets(str);

printf("Enter a character :");

scanf("%c",&a);

printf("%d\n",f(str,a));

return 0;

}

int f(char \*s,char a)

{

if(\*s=='\0')

return 0;

if(\*s==a)

return 1 + f(s+1,a);

return f(s+1,a);

}

12.

#include<stdio.h>

void func1(int n);

void func2(int n);

int main(void)

{

func1(4);

printf("\n");

func2(4);

return 0;

}

void func1(int n)

{

int i;

if(n==0)

return;

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

printf("\*");

printf("\n");

func1(n-1);

}

void func2(int n)

{

int i;

if(n==0)

return;

func2(n-1);

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

printf("\*");

printf("\n");

}

13.

#include<stdio.h>

int InputAndAdd(int n);

int main(void)

{

int n;

printf("Enter n :");

scanf("%d",&n);

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

return 0;

}

int InputAndAdd(int n)

{

int a;

printf("Enter a number : ");

scanf("%d",&a);

if(n==1)

return a;

return a + InputAndAdd(n-1);

}

14.

#include<stdio.h>

void func(void);

int main(void)

{

printf("Enter text :\n");

func();

printf("\n");

return 0;

}/\*End of main()\*/

void func(void)

{

char c;

if((c=getchar())!='\n')

func();

putchar(c);

}

15.

#include<stdio.h>

#include<math.h>

int countPrimes(int a, int b);

int isprime(int n);

int main(void)

{

int a,b;

printf("Enter values of a and b :");

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

printf("\nTotal prime numbers = %d\n",countPrimes(a,b));

return 0;

}

countPrimes(int a, int b)

{

if(a>b)

return 0;

if(isprime(b))

{

printf("%d ",b);

return 1 + countPrimes(a,b-1);

}

else

return countPrimes(a,b-1);

}

int isprime(int n)

{

int i;

for(i=2; i<=sqrt(n); i++)

if(n%i == 0)

return 0;

return 1;

}

16.

#include<stdio.h>

int sumdiv(int num, int x);

int main(void)

{

int num;

printf("Enter a number : ");

scanf("%d",&num);

printf("\nSum of divisors = %d\n",sumdiv(num,num/2));

return 0;

}

sumdiv(int num, int x)

{

if(x==1)

{

printf("%d ",x);

return 1;

}

if(num%x==0)/\*if x is a proper divisor\*/

{

printf("%d ",x);

return x + sumdiv(num,x-1);

}

else

return sumdiv(num,x-1);

}

17.

#include<stdio.h>

int sumdiv(int num, int x);

int main(void)

{

int num;

printf("Enter a number :");

scanf("%d",&num);

if(sumdiv(num, num/2) == num)

printf("Perfect\n");

else

printf("Not Perfect\n");

return 0;

}

sumdiv(int num, int x)

{

if(x==1)

return 1;

if(num%x==0)/\*if x is a proper divisor\*/

return x + sumdiv(num,x-1);

else

return sumdiv(num,x-1);

}

18.

#include<stdio.h>

void f(int n);

int main(void)

{

int num=12340;

f(num);

return 0;

}

void f(int n)

{

if(n==0)

return;

f(n/10);

switch(n%10)

{

case 0: printf("zero ");break;

case 1: printf("one ");break;

case 2: printf("two ");break;

case 3: printf("three ");break;

case 4: printf("four ");break;

case 5: printf("five ");break;

case 6: printf("six ");break;

case 7: printf("seven ");break;

case 8: printf("eight ");break;

case 9: printf("nine ");break;

}

}

19.

#include<stdio.h>

void reverse(int n, int \*p\_rev);

int main(void)

{

int rev=0;

reverse(1234,&rev);

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

rev=0;

reverse(4567,&rev);

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

return 0;

}

void reverse(int n, int \*p\_rev)

{

if(n==0)

return;

\*p\_rev = \*p\_rev \* 10 + n%10;

reverse(n/10, p\_rev);

}

20.

#include<stdio.h>

int rem(int a,int b);

int main(void)

{

int a,b;

printf("Enter two numbers :");

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

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

return 0;

}

int rem(int a,int b)

{

if(a<b)

return a;

return rem(a-b,b);

}

21.

#include<stdio.h>

int quo(int a,int b);

int main(void)

{

int a,b;

printf("Enter two numbers :");

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

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

return 0;

}

int quo(int a,int b)

{

if(a<b)

return 0;

return 1 + quo(a-b,b);

}

22.

#include<stdio.h>

int main(void)

{

int a,b;

printf("Enter two numbers :");

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

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

return 0;

}

int pow(int a, int n)

{

if (n==0)

return 1;

else if (n%2==0)

return pow(a\*a, n/2);

else

return a \* pow(a\*a, (n-1)/2);

}

23.

#include<stdio.h>

int f(int a,int b);

int main(void)

{

int a,b;

printf("Enter two numbers : ");

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

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

return 0;

}

int f(int a,int b)

{

if(a==0) /\*if we write if(a==1) return b; then 0 \* b can not be computed, so this condition\*/

return 0;

if(a%2!=0) /\*if a is odd\*/

return b + f(a/2, b\*2);

return f(a/2, b\*2);

}

24.

#include<stdio.h>

int log2(int num);

int logN(int num,int base);

int main(void)

{

int num, base;

printf("Enter a number :");

scanf("%d",&num);

printf("%d\n",log2(num));

printf("Enter a number and a base :");

scanf("%d%d",&num,&base);

printf("%d\n",logN(num,base)) ;

return 0;

}

int log2(int num)

{

if(num==1)

return 0;

return 1 + log2(num/2);

}

int logN(int num,int base)

{

if(num<base)

return 0;

return 1 + logN(num/base,base);

}

25.

#include<stdio.h>

int BC(int n, int k);

int main(void)

{

int n,k;

printf("Enter n and k : ");

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

printf("%d\n",BC(n,k));

return 0;

}

int BC(int n,int k)

{

if(k==0 || k==n)

return 1;

return BC(n-1,k-1) + BC(n-1,k);

}

26.

#include<stdio.h>

long int fact(int num);

double power(float x,int n);

double series(float x,int n);

double rseries(float x,int n);

int main(void)

{

float x;

int n;

printf("Enter x : ");

scanf("%f", &x);

printf("Enter number of terms : ");

scanf("%d", &n);

printf("Iterative %lf\n",series(x,n));

printf("Recursive %lf\n",rseries(x,n));

return 0;

}

long int fact(int num)

{

int i;

long int f=1;

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

f=f\*i;

return f;

}

double power(float x,int n)

{

int i;

float p=1;

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

p = p\*x;

return p;

}

double series(float x,int n)

{

int i,j,sign=1;

float term,sum=0;

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

{

sign = (i%2==0)?-1:1;

j = 2\*i-1;

term = sign\*power(x,j)/fact(j);

sum+=term;

}

return sum;

}

double rseries(float x,int n)

{

int sign=1;

float term;

if(n==0)

return 0;

sign = (n%2==0)?-1:1;

term = sign \* power(x,2\*n-1)/fact(2\*n-1);

return term + rseries(x,n-1);

}

27.

#include<stdio.h>

long int fact(int num);

double power(float x,int n);

double series(float x,int n);

double rseries(float x,int n);

int main(void)

{

float x;

int n;

printf("Enter x : ");

scanf("%f", &x);

printf("Enter number of terms : ");

scanf("%d", &n);

printf("Iterative %lf\n",series(x,n));

printf("Recursive %lf\n",rseries(x,n));

return 0;

}

long int fact(int num)

{

int i;

long int f=1;

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

f=f\*i;

return f;

}

double power(float x,int n)

{

int i;

float p=1;

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

p = p\*x;

return p;

}

double series(float x,int n)

{

int i,j,sign=1;

float term,sum=0;

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

{

sign = (i%2==0)?-1:1;

j = 2\*i-1;

term = sign\*power(x,j)/fact(j);

sum+=term;

}

return sum;

}

double rseries(float x,int n)

{

int sign=1;

float term;

if(n==0)

return 0;

sign = (n%2==0)?-1:1;

term = sign \* power(x,2\*n-1)/fact(2\*n-1);

return term + rseries(x,n-1);

}

28.

#include<stdio.h>

#include<math.h>

int func(int n);

int main(void)

{

int n;

printf("Enter n :");

scanf("%d",&n);

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

return 0;

}

func(int n)

{

if(n==1)

return 1;

return n + func(n-1);

}

Execise-Arrays

1.

#include<stdio.h>

int main(void)

{

int i,size=5,arr[size];

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

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

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

printf("%d ",arr[i]);

return 0;

}

2.

#include<stdio.h>

int main(void)

{

int arr[4]={2,4,8,16},i=3,j=0;

while(i)

{

j+=arr[i];

i--;

}

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

return 0;

}

3.

#include<stdio.h>

int main(void)

{

int i=0,j=0,arr[6]={4,2,6,0,5,10};

while(arr[i])

{

j+=arr[i];

i++;

}

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

return 0;

}

4.

#include<stdio.h>

void func(int arr[]);

int main(void)

{

int arr[5]={5,10,15,20,25};

func(arr);

return 0;

}

void func(int arr[])

{

int i=5,sum=0;

while(i>2)

sum=sum+arr[--i];

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

}

5.

#include<stdio.h>

int main(void)

{

int x[10],y[3][4],z[2][3][5];

printf("%u\t%u\t%u\n",sizeof(x),sizeof(y),sizeof(z));

return 0;

}

6.

#include<stdio.h>

void swapvar(int a,int b);

void swaparr(int arr1[5],int arr2[5]);

int main(void)

{

int a=4,b=6;

int arr1[5]={1,2,3,4,5};

int arr2[5]={6,7,8,9,10};

swapvar(a,b);

swaparr(arr1,arr2);

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

printf("arr1[0]=%d, arr1[4]=%d\n",arr1[0],arr1[4]);

printf("arr2[0]=%d, arr2[4]=%d\n",arr2[0],arr2[4]);

return 0;

}

void swapvar(int a,int b)

{

int temp;

temp=a, a=b, b=temp;

}

void swaparr(int arr1[5],int arr2[5])

{

int i,temp;

for(i=0; i<5; i++)

{ temp=arr1[i], arr1[i]=arr2[i], arr2[i]=temp; }

}

7.

#include<stdio.h>

int main(void)

{

int i,j,arr[3][4]={ {1,2,3,4}, {5,6,7,8}, {9,10,11,12} };

for(i=0; i<4; i++)

{

for(j=0; j<3; j++)

printf("%3d",arr[j][i]);

printf("\n");

}

return 0;

}

8.

#include<stdio.h>

int main(void)

{

int a[10]={2,-3,4,-5,6,7,1,9,-10,-11};

int i,j,x,k=0;

for(i=0; i<10; i++)

{

x=a[k];

if(x<0)

{

for(j=k; j<10; j++)

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

a[9]=x;

}

else

k++;

}

for(i=0; i<10; i++)

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

printf("\n");

return 0;

}

9.

#include<stdio.h>

#define N 10

int main(void)

{

int i,a[N];

a[0]=a[1]=1;

for(i=2; i<N; i++)

a[i]=a[i-1]+a[i-2];

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

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

return 0;

}

10.

#include<stdio.h>

#define N 6

int main(void)

{

int i,j,a[N]={1,2,3,4,5,6};

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

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

a[i]+=a[j];

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

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

return 0;

}

11.

#include<stdio.h>

#define N 6

int main(void)

{

int i,j,a[N]={1,2,3,4,5,6};

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

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

a[i]+=a[j];

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

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

return 0;

}

12.

#include<stdio.h>

#define N 10

int main(void)

{

int i,j,a[N]={1};

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

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

a[i]+=a[j];

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

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

return 0;

}

13.

#include<stdio.h>

int func(int arr[],int size);

int main(void)

{

int arr[10]={1,2,3,4,8,10};

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

return 0;

}

int func(int arr[],int size)

{

if(size==0)

return 0;

else if(arr[size-1]%2==0)

return 1 + func(arr,size-1);

else

return func(arr,size-1);

}

14.

#include<stdio.h>

int func(int arr[],int size);

int main(void)

{

int arr[10]={1,2,3,4,8,10};

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

return 0;

}

int func(int arr[],int size)

{

if(size==0)

return 0;

else if(arr[size-1]%2==0)

return 1 + func(arr,size-1);

else

return func(arr,size-1);

}

15.

#include<stdio.h>

int main(void)

{

int arr[10]={1,2,3,4,8,10};

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

return 0;

}

int func(int arr[],int size)

{

if(size==0)

return 0;

return arr[size-1] + func(arr,size-1);

}

16.

#include<stdio.h>

int func(int arr[],int size);

int main(void)

{

int arr[10]={2,3,1,4,6,34};

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

return 0;

}

int func(int arr[],int size)

{

int m;

if(size==1)

return arr[0];

m=func(arr,size-1);

if(arr[size-1] < m)

return arr[size-1];

else

return m;

}

17.

#include<stdio.h>

int func(int arr[],int low,int high);

int main(void)

{

int arr[10]={3,4,2,11,8,10};

printf("%d\n",func(arr,0,5));

return 0;

}

int func(int arr[],int low,int high)

{

int mid,left,right;

if(low==high)

return arr[low];

mid=(low+high)/2;

left=func(arr,low,mid);

right=func(arr,mid+1,high);

if(left<right)

return left;

else

return right;

}

18.

#include<stdio.h>

#define MAX 50

int main(void)

{

int i,arr[MAX],size,temp;

printf("Enter size of array : ");

scanf("%d",&size);

printf("Enter array : ");

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

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

for(i=0; i<size-1; i=i+2)

{

temp=arr[i];

arr[i]=arr[i+1];

arr[i+1]=temp;

}

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

printf("%d ",arr[i]);

return 0;

}

19.

#include <stdio.h>

#define MAX 50

int main(void)

{

int i,arr[MAX],size,max,min;

printf("Enter size of array : ");

scanf("%d",&size);

printf("Enter array : ");

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

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

max=min=arr[0];

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

{

if(arr[i]>max)

max=arr[i];

if(arr[i]<min)

min=arr[i];

}

printf("%d %d %d\n",max,min,max-min);

return 0;

}

20.

#include <stdio.h>

#include<limits.h>

#include<stdlib.h>

#define MAX 50

int main(void)

{

int i,arr[MAX],size,max,smax;

printf("Enter size of array : ");

scanf("%d",&size);

printf("Enter array : ");

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

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

if(size<2)

{

printf("There should be at least two elements\n");

exit(1);

}

max=smax=INT\_MIN; /\*Include limits.h for INT\_MIN\*/

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

if(arr[i] > max)

{

smax=max;

max=arr[i];

}/\*If arr[i] is less than max but greater than smax, change only smax\*/

else if(arr[i] > smax)

smax = arr[i];

printf("Largest=%d, Second Largest=%d\n",max,smax);

return 0;

}

21.

#include <stdio.h>

#define MAX 50

int main(void)

{

int i,arr[MAX],size,j,n,temp;

printf("Enter size of array : ");

scanf("%d",&size);

printf("Enter array : ");

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

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

printf("Enter n : ");

scanf("%d",&n);

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

{

temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

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

printf("%d ",arr[i]);

return 0;

}

22.

#include <stdio.h>

#define MAX 100

int main(void)

{

int arr[MAX],i,j,temp,n,xchanges;

printf("Enter the number of elements : ");

scanf("%d",&n);

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

{

printf("Enter element %d : ",i+1);

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

}

/\*Bubble sort\*/

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

{

xchanges = 0;

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

{

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

{

temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

xchanges++;

}

}

if(xchanges==0) /\*If list is sorted\*/

break;

}

printf("List in Ascending order is :\n");

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

printf("%d ",arr[i]);

printf("\n");

printf("List in Descending order is :\n");

for(i=n-1; i>=0; i--)

printf("%d ",arr[i]);

printf("\n");

return 0;

}

23.

#include <stdio.h>

#define MAX 100

void selection(int arr[], int n);

void bubble(int arr[], int n);

void insertion(int arr[], int n);

int main(void)

{

int i,n=10;

int arr1[MAX]={4,5,7,9,8,3,1,5,6,2};

int arr2[MAX]={4,5,7,9,8,3,1,5,6,2};

int arr3[MAX]={4,5,7,9,8,3,1,5,6,2};

selection(arr1,n);

bubble(arr2,n);

insertion(arr3,n);

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

printf("%d ",arr1[i]);

printf("\n");

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

printf("%d ",arr2[i]);

printf("\n");

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

printf("%d ",arr3[i]);

printf("\n");

return 0;

}/\*End of main()\*/

void selection(int arr[],int n)

{

int i,j,max,temp;

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

{

max = i;

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

{

if(arr[max] < arr[j])

max = j ;

}

if(i!=max)

{

temp = arr[i];

arr[i] = arr[max];

arr[max] = temp ;

}

}

}

void bubble(int arr[],int n)

{

int i,j,xchanges,temp;

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

{

xchanges = 0;

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

{

if(arr[j] < arr[j+1])

{

temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

xchanges++;

}

}

if(xchanges==0) /\*If list is sorted\*/

break;

}

}

void insertion(int arr[],int n)

{

int i,j,k;

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

{

k=arr[i];

for(j=i-1; j>=0 && k>arr[j]; j--)

arr[j+1]=arr[j];

arr[j+1]=k;

}

}

24.

#include <stdio.h>

#define MAX 50

int AllDistinct(int arr[],int size);

int main(void)

{

int i,arr[MAX],size;

printf("Enter size of array : ");

scanf("%d",&size);

printf("Enter array : ");

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

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

if(AllDistinct(arr,size))

printf("All elements are distinct\n");

else

printf("All elements are not distinct\n");

return 0;

}

int AllDistinct(int arr[],int size)

{

int i,j;

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

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

if(arr[i]==arr[j])/\*Duplicate found\*/

return 0;

return 1;

}

25.

#include <stdio.h>

#define MAX 50

int deleteDuplicates(int arr[],int size);

int main(void)

{

int i,arr[MAX],size;

printf("Enter size of array : ");

scanf("%d",&size);

printf("Enter array : ");

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

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

size=deleteDuplicates(arr,size);

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

printf("%d ",arr[i]);

return 0;

}

int deleteDuplicates(int arr[],int size)

{

int i,j,k;

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

{

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

{

if(arr[i]==arr[j])/\*Duplicate found\*/

{

size--;

for(k=j; k<size; k++)

arr[k]=arr[k+1];

}

else

j++; /\*Continue Comparing\*/

}

}

return size;/\*Return the new size of the array\*/

}

26.

#include <stdio.h>

#define MAX 50

int deleteDuplicates(int arr[],int size);

int main(void)

{

int i,arr[MAX],size;

printf("Enter size of array : ");

scanf("%d",&size);

printf("Enter array : ");

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

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

size=deleteDuplicates(arr,size);

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

printf("%d ",arr[i]);

return 0;

}

int deleteDuplicates(int arr[],int size)

{

int i,k,n\_dup;

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

{

if(arr[i]==arr[i+1])

{

k=i+1;

while(k<size && arr[k]==arr[k+1])

k++;

/\*Duplicates of arr[i] start at i+1 and end at k\*/

n\_dup=k-i; /\*Number of duplictaes of arr[i]\*/

for(k=i+1; k<size; k++)

arr[k]=arr[k+n\_dup];

size=size-n\_dup;

}

}

return size;/\*Return the new size of the array\*/

}

27.  
#include <stdio.h>

#define MAX 50

int main(void)

{

int i,j,n,arr[MAX],inv=0;

printf("Enter size of array : ");

scanf("%d",&n);

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

{

printf("%d : ",i);

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

}

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

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

if(arr[i] > arr[j])

inv++;

printf("Total inversions in the array : %d\n",inv);

return 0;

}

28.

#include <stdio.h>

#define MAX 50

int main(void)

{

int i,size,arr[MAX],count;

int max\_freq=0;

int mf\_element,element;

printf("Enter the number of elements : ");

scanf("%d",&size);

printf("Enter the elements in sorted order : \n");

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

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

mf\_element=arr[0];

max\_freq=0;

i=0;

while(i<=size-1)

{

element=arr[i];

count=0;

while(arr[i]==element && i<size)

{

count++;

i++;

}

if(count > max\_freq)

{

mf\_element=element;

max\_freq=count;

}

}

printf("%d occurs %d times\n", mf\_element,max\_freq);

return 0;

}

29.

#include <stdio.h>

#define MAX 50

int main(void)

{

int i,size,arr[MAX],leader;

printf("Enter size of array :");

scanf("%d",&size);

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

{

printf("%d : ",i);

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

}

leader=arr[size-1]; /\*Last element is a leader\*/

printf("%d ", leader);

for(i=size-2; i>=0; i--)

if(arr[i]>leader)

{

/\*A new leader is found\*/

leader=arr[i];

printf("%d ",leader);

}

return 0;

}

30.

#include <stdio.h>

#define MAX 50

int LastOcc(int arr[],int size,int item);

int firstOcc(int arr[],int size,int item);

int main(void)

{

int i,size,item,arr[MAX],index;

printf("Enter the number of elements : ");

scanf("%d",&size);

printf("Enter the elements(in sorted order) : \n");

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

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

printf("Enter the item to be searched : ");

scanf("%d",&item);

index=firstOcc(arr,size,item);

if(index==-1)

printf("%d not found in array\n",item);

else

{

printf("First occurence of %d is at position %d\n",item,index);

index=LastOcc(arr,size,item);

printf("Last occurence of %d is at position %d\n",item,index);

}

return 0;

}

int firstOcc(int arr[],int size,int item)

{

int low=0,up=size-1,mid;

if(arr[0]==item)

return 0;

while(low<=up)

{

mid=(low+up)/2;

if(arr[mid-1]<item && arr[mid]==item)

return mid;

if(item>arr[mid])

low=mid+1; /\*Search in right half\*/

else if(item<=arr[mid]) /\*if equal we'll search in left half\*/

up=mid-1; /\*Search in left half\*/

}

return -1;

}

int LastOcc(int arr[],int size,int item)

{

int low=0,up=size-1,mid;

if(arr[size-1]==item)

return size-1;

while(low<=up)

{

mid=(low+up)/2;

if(arr[mid+1]>item && arr[mid]==item)

return mid;

if(item>=arr[mid])/\*if equal we'll search in right half \*/

low=mid+1; /\*Search in right half \*/

else if(item < arr[mid])

up=mid-1; /\*Search in left half \*/

}

return -1;

}

31.

#include<stdio.h>

#define MAX 20

void sort\_Columnwise(int a[MAX][MAX],int m,int n);

void sort\_RowWise(int a[MAX][MAX],int m,int n);

int main(void)

{

int a[MAX][MAX],i,j,m,n;

printf("Enter number of rows : ");

scanf("%d",&m);

printf("Enter number of columns : ");

scanf("%d",&n);

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

{

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

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

printf("\n");

}

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

{

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

printf("%4d",a[i][j]);

printf("\n");

}

sort\_RowWise(a,m,n);

printf("\n");

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

{

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

printf("%4d",a[i][j]);

printf("\n");

}

sort\_Columnwise(a,m,n);

printf("\n");

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

{

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

printf("%4d",a[i][j]);

printf("\n");

}

return 0;

}

void sort\_Columnwise(int a[MAX][MAX],int m,int n)

{

int k,xchanges,i,j,temp;

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

{

/\*Bubble sort\*/

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

{

xchanges=0;

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

{

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

{

temp=a[j][k];

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

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

xchanges++;

}

}

if(xchanges==0) /\*If list is sorted\*/

break;

}

}

}

32.

#include<stdio.h>

#define MAX 20

void sort\_Rowwise(int a[MAX][MAX],int m,int n);

void sort\_Columnwise(int a[MAX][MAX],int m,int n);

int search(int a[MAX][MAX],int m,int n,int item);

int main(void)

{

int a[MAX][MAX],i,j,m,n,item;

printf("Enter the number of rows : ");

scanf("%d",&m);

printf("Enter the number of columns : ");

scanf("%d",&n);

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

{

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

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

printf("\n");

}

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

{

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

printf("%4d",a[i][j]);

printf("\n");

}

printf("\n");

sort\_Rowwise(a,m,n);

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

{

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

printf("%4d",a[i][j]);

printf("\n");

}

sort\_Columnwise(a,m,n);

printf("\n");

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

{

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

printf("%4d",a[i][j]);

printf("\n");

}

printf("Enter item to be searched: ");

scanf("%d",&item);

if(!search(a,m,n,item))

printf("Not found\n");

return 0;

}

int search(int a[MAX][MAX], int m, int n, int item)

{

int i,j;

i=0,j=n-1;

while(i<=n-1 && j>=0)

{

if(a[i][j] == item)

{

printf("Found at Row:%d, Column:%d\n", i,j);

return 1;

}

if( a[i][j] < item )

i++;

else

j--;

}

return 0;

}

void sort\_Rowwise(int a[MAX][MAX], int m, int n)

{

int i,j,k, temp,xchanges;

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

{

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

{

xchanges = 0;

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

{

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

{

temp = a[k][j];

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

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

xchanges++;

}

}

if(xchanges==0) /\*If list is sorted\*/

break;

}

}

}

void sort\_Columnwise(int a[MAX][MAX], int m, int n)

{

int k,xchanges,i,j,temp;

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

{

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

{

xchanges = 0;

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

{

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

{

temp = a[j][k];

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

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

xchanges++;

}

}

if(xchanges==0) /\*If list is sorted\*/

break;

}

}

}

33.

#include<stdio.h>

#define MAX 20

int main(void)

{

int a[MAX][MAX],i,j,m,n,temp;

printf("Enter the number of rows : ");

scanf("%d",&m);

printf("Enter the number of columns : ");

scanf("%d",&n);

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

{

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

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

printf("\n");

}

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

{

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

printf("%4d",a[i][j]);

printf("\n");

}

/\*Interchange rows 0 and m-1\*/

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

{

temp=a[0][j];

a[0][j]=a[m-1][j];

a[m-1][j]=temp;

}

printf("\n\n");

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

{

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

printf("%4d",a[i][j]);

printf("\n");

}

return 0;

}

34.

#include<stdio.h>

#define MAX 20

int main(void)

{

int a[MAX][MAX],i,j,m,n,temp,p,q;

printf("Enter the number of rows : ");

scanf("%d",&m);

printf("Enter the number of columns : ");

scanf("%d",&n);

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

{

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

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

printf("\n");

}

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

{

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

printf("%4d",a[i][j]);

printf("\n");

}

for(p=0,q=m-1; p<q; p++,q--)

/\*Interchange rows p and q\*/

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

{

temp=a[p][j];

a[p][j]=a[q][j];

a[q][j]=temp;

}

printf("\n");

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

{

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

printf("%4d",a[i][j]);

printf("\n");

}

return 0;

}

35.

#include<stdio.h>

#define MAX 20

int main(void)

{

int a[MAX][MAX],i,j,m,n,temp;

printf("Enter the number of rows : ");

scanf("%d",&m);

printf("Enter the number of columns : ");

scanf("%d",&n);

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

{

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

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

printf("\n");

}

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

{

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

printf("%4d",a[i][j]);

printf("\n");

}

/\*Interchange columns 0 and n-1\*/

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

{

temp=a[i][0];

a[i][0]=a[i][n-1];

a[i][n-1]=temp;

}

printf("\n");

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

{

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

printf("%4d",a[i][j]);

printf("\n");

}

return 0;

}

36.

#include<stdio.h>

#define MAX 20

int main(void)

{

int a[MAX][MAX],i,j,m,n,temp,p,q;

printf("Enter the number of rows : ");

scanf("%d",&m);

printf("Enter the number of columns : ");

scanf("%d",&n);

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

{

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

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

printf("\n");

}

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

{

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

printf("%4d",a[i][j]);

printf("\n");

}

for(p=0,q=n-1; p<q; p++,q--)

/\*Interchange columns p and q\*/

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

{

temp=a[i][p];

a[i][p]=a[i][q];

a[i][q]=temp;

}

printf("\n");

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

{

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

printf("%4d",a[i][j]);

printf("\n");

}

return 0;

}

37.

#include<stdio.h>

#define MAX 20

int isSymmetric(int a[MAX][MAX],int n);

int main(void)

{

int a[MAX][MAX],i,j,n;

printf("Enter the number of rows : ");

scanf("%d",&n);

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

{

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

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

printf("\n");

}

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

{

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

printf("%4d",a[i][j]);

printf("\n");

}

if(isSymmetric(a,n))

printf("Matrix is symmetric\n");

else

printf("Matrix is not symmetric\n");

return 0;

}

int isSymmetric(int a[MAX][MAX],int n)

{

int i,j;

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

{

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

if( a[i][j] != a[j][i] )

return 0;

}

return 1;

}

38.

#include<stdio.h>

int sumEven(int arr[],int size);

int main(void)

{

int arr[6]={1,2,3,4,8,10};

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

return 0;

}

int sumEven(int arr[],int size)

{

if(size==0)

return 0;

else if(arr[size-1]%2==0)

return arr[size-1] + sumEven(arr,size-1);

else

return sumEven(arr,size-1);

}

39.

#include<stdio.h>

int func(int arr[],int low,int high);

int main(void)

{

int arr[6]={1,2,3,4,8,10};

printf("%d\n",func(arr,0,5));

return 0;

}

int func(int arr[],int low,int high)

{

int mid, left, right;

if(low==high)

return arr[low];

mid=(low+high)/2;

left=func(arr,low,mid);

right=func(arr,mid+1,high);

return left+right;

}

40.

#include<stdio.h>

void reverse(int arr[],int low,int high);

int main(void)

{

int i,arr[6]={1,2,3,4,8,10};

reverse(arr,0,5);

for(i=0; i<=5; i++)

printf("%d ",arr[i]);

return 0;

}

void reverse(int arr[],int low,int high)

{

int tmp;

if(low>=high)

return;

tmp=arr[low];

arr[low]=arr[high];

arr[high]=tmp;

reverse(arr,low+1,high-1);

}

41.

#include<stdio.h>

int isAscending(int arr[],int size);

int main(void)

{

int i,n,arr[50];

printf("Enter number of elements :");

scanf("%d",&n);

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

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

if(isAscending(arr,n))

printf("Array elements are in strict ascending order\n");

else

printf("Array elements are not in strict ascending order\n");

return 0;

}

int isAscending(int arr[],int size)

{

if(size==1)

return 1;

if(arr[0]>=arr[1])

return 0;

return isAscending(arr+1,size-1);

}

42.

#include<stdio.h>

#define MAX 20

int main(void)

{

int a[MAX][MAX],i,j,m,n;

printf("Enter value of m (rows): ");

scanf("%d",&m);

printf("Enter value of n (columns): ");

scanf("%d",&n);

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

{

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

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

printf("\n");

}

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

{

a[i][n]=0;

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

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

}

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

{

a[m][j]=0;

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

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

}

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

{

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

printf("%4d",a[i][j]);

printf("\n");

}

return 0;

}

43.

#include <stdio.h>

#define MAX 20

int main(void)

{

int a[MAX][MAX],i,j,k,m,n,rStart,cStart,rEnd,cEnd;

printf("Enter value of m (rows): ");

scanf("%d", &m);

printf("Enter value of n (columns): ");

scanf("%d", &n);

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

{

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

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

printf("\n");

}

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

{

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

printf("%4d", a[i][j]);

printf("\n");

}

printf("\n\n");

for(rStart=0,cStart=0,rEnd=m-1,cEnd=n-1; rStart<=rEnd && cStart<=cEnd; rStart++,cStart++,rEnd--,cEnd--)

{

for(k=cStart; k<=cEnd; k++)

printf("%d\* ",a[rStart][k]);

for(k=rStart+1; k<=rEnd; k++)

printf("%d$ ",a[k][cEnd]);

if(rStart<rEnd)

for(k=cEnd-1; k>=cStart; k--)

printf("%d# ",a[rEnd][k]);

if(cStart<cEnd)

for(k=rEnd-1; k>=rStart+1; k--)

printf("%d& ",a[k][cStart]);

}

return 0;

}

44.

#include<stdio.h>

#define MAX 20

int main(void)

{

int a[MAX][MAX],i,j,n,start,end;

printf("Enter value of n : ");

scanf("%d", &n);

j=1;

for(start=0,end=n-1; start<=end; start++,end--)

{

for(i=start; i<=end; i++)

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

for(i=start+1; i<=end; i++)

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

for(i=end-1; i>=start; i--)

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

for(i=end-1; i>=start+1; i--)

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

}

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

{

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

printf("%4d", a[i][j]);

printf("\n");

}

printf("\n\n");

return 0;

}

45.

#include<stdio.h>

#define MAX 50

void partition(int a[],int size);

int main(void)

{

int i,arr[MAX],size;

printf("Enter size of array :");

scanf("%d",&size);

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

{

printf("%d : ",i);

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

}

partition(arr,size);

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

printf("%d ",arr[i]);

return 0;

}

void partition(int a[],int size)

{

int i=0,j=size-1,temp;

while(i<j)

{

while(a[i]<0 && i<j) /\*Move right\*/

i++;

while(a[j]>=0 && i<j) /\*Move left\*/

j--;

if(i<j)

{

temp=a[i];

a[i]=a[j];

a[j]=temp;

i++; j--;

}

}

}

46.

#include<stdio.h>

#define MAX 50

int main(void)

{

int i,j,size,arr[MAX],next\_ge[MAX];

printf("Enter size of array :");

scanf("%d",&size);

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

{

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

next\_ge[i]=-1;

}

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

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

if(arr[i] < arr[j])

{

next\_ge[i] = arr[j];

break;

}

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

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

printf("\n");

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

printf("%d\t",next\_ge[i]);

return 0;

}#include<stdio.h>

#define MAX 50

int main(void)

{

int i,j,size,arr[MAX],next\_ge[MAX];

printf("Enter size of array :");

scanf("%d",&size);

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

{

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

next\_ge[i]=-1;

}

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

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

if(arr[i] < arr[j])

{

next\_ge[i] = arr[j];

break;

}

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

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

printf("\n");

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

printf("%d\t",next\_ge[i]);

return 0;

}

47.

#include<stdio.h>

#define MAX 50

int main(void)

{

int i,j,n,arr[MAX],min,temp;

int k=3;

printf("Enter size of array :");

scanf("%d",&n);

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

{

printf("%d : ",i);

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

}

for(i=0; i<k; i++)/\*Selection sort upto k only\*/

{

min=i;

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

{

if(arr[min] > arr[j])

min=j ;

}

if(i!=min)

{

temp=arr[i];

arr[i]=arr[min];

arr[min]=temp;

}

}

printf("kth smallest element is %d\n",arr[k-1]);

return 0;

}

48.

#include<stdio.h>

void reverse(int a[], int start, int end);

int main(void)

{

int a[10]={0,1,2,3,4,5,6,7,8,9},i;

reverse(a,4,7);

for(i=0; i<10; i++)

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

return 0;

}

void reverse(int a[], int start, int end)

{

int i,j,temp;

for(i=start,j=end; i<j; i++,j--)

{

temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

49.

#include<stdio.h>

#define MAX 50

int main(void)

{

int i,arr[MAX],temp,size;

printf("Enter size of array : ");

scanf("%d",&size);

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

{

printf("%d : ",i);

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

}

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

printf("%d ",arr[i]);

printf("\n");

temp=arr[0];

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

arr[i-1]=arr[i];

arr[size-1]=temp;

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

printf("%d ",arr[i]);

return 0;

}

50.

#include<stdio.h>

#define MAX 50

void rotate1(int a[],int size,int k);

void rotate2(int a[], int size, int k);

void reverse(int a[], int start, int end);

int main(void)

{

int a[MAX],size,i,k;

printf("Enter size of array : ");

scanf("%d",&size);

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

{

printf("%d : ",i);

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

}

printf("Enter k(How many times to rotate left): ");

scanf("%d",&k);

rotate1(a,size,k);

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

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

printf("\n");

rotate2(a,size,k);

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

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

return 0;

}

/\*rotate left by 1,k times\*/

void rotate1(int a[],int size,int k)

{

int i,j,temp;

for(j=1; j<=k; j++)

{

temp=a[0];

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

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

a[size-1]=temp;

}

}

/\*rotate left by k using reverse()\*/

void rotate2(int a[], int size, int k)

{

reverse(a,0,k-1);

reverse(a,k,size-1);

reverse(a,0,size-1);

}

void reverse(int a[], int start, int end)

{

int i,j,temp;

for(i=start,j=end; i<j; i++,j--)

{

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

Exercises-Pointers 9

1.

#include<stdio.h>

int main(void)

{

int a=5,\*ptr;

ptr=&a;

printf("Input a number : ");

scanf("%d",ptr); /\*Suppose the input number is 16\*/

printf("%d %d\n",a,\*ptr);

return 0;

}

2.

#include<stdio.h>

int main(void)

{

int \*ptr;

printf("Enter a number : ");

scanf("%d",ptr);

printf("%d\n",\*ptr);

return 0;

}

3.

#include<stdio.h>

int main(void)

{

int arr[5],i;

for(i=0; i<5; i++)

printf("%p ",arr+i);

printf("\nEnter 5 numbers : ");

for(i=0; i<5; i++)

scanf("%d",arr+i);

for(i=0; i<5; i++)

printf("%d ",\*(arr+i));

return 0;

}

4.

#include<stdio.h>

int main(void)

{

int i,arr[5]={25,30,35,40,45},\*p;

p=arr;

for(i=0; i<5; i++)

printf("%d\t%d\t",\*(p+i),p[i]);

return 0;

}

5.

#include<stdio.h>

int main(void)

{

int i,arr[5]= {25,30,35,40,45},\*p;

p=&arr[4];

for(i=0; i<5; i++)

printf("%d\t%d\t",\*(p-i),p[-i]);

return 0;

}

6.

#include<stdio.h>

int main(void)

{

int i,arr[5] = {25,30,35,40,55},\*p;

for(i=0; i<5; i++)

{

printf("%d ",\*arr);

arr++;

}

return 0;

}

7.

#include<stdio.h>

int main(void)

{

int i,arr[5]={25,30,35,40,45},\*p=arr;

for(i=0; i<5; i++)

{

(\*p)++;

printf("%d ",\*p);

p++;

}

return 0;

}

8.

#include<stdio.h>

int main(void)

{

int i,arr[5]={25,40,55,70,85},\*p=arr;

for(i=0; i<5; i++)

printf("%d ",\*p++);

printf("\n");

for(i=0; i<5; i++)

printf("%d ",\*--p);

printf("\n");

return 0;

}

9.

#include<stdio.h>

int main(void)

{

int i,arr[5]={25,40,55,70,85},\*p=arr;

for(i=0; i<8; i++)

printf("%d ",++\*p);

printf("\n");

for(i=0; i<7; i++)

printf("%d ",(\*p)++);

printf("\n");

return 0;

}

10.

#include<stdio.h>

int main(void)

{

int arr[10]={25,30,35,40,55,60,65,70,85,90},\*p;

for(p=&arr[0]; p<arr+10; p++)

printf("%d ",\*p);

return 0;

}

11.

#include<stdio.h>

int main(void)

{

int arr[10]= {25,30,35,40,55,60,65,70,85,90},\*p;

for(p=arr+2; p<arr+8; p=p+2)

printf("%d ",\*p);

return 0;

}

12.

#include<stdio.h>

int main(void)

{

int i,arr[10]={25,30,35,40,55,60,65,70,85,90};

int \*p=arr+9;

for(i=0; i<10; i++)

printf("%d ",\*p--);

return 0;

}

13.

#include<stdio.h>

int main(void)

{

int arr[10]={25,30,35,40,55,60,65,70,85,90},\*p;

for(p=arr+9; p>=arr; p--)

printf("%d ",\*p);

return 0;

}

14.

#include<stdio.h>

int main(void)

{

int arr[4]={10,20,30,40};

int x=100, \*ptr=arr;

printf("%p %d %d\n",ptr,\*ptr,x);

x=\*ptr++;

printf("%p %d %d\n",ptr,\*ptr,x);

x=\*++ptr;

printf("%p %d %d\n",ptr,\*ptr,x);

x=++\*ptr;

printf("%p %d %d\n",ptr,\*ptr,x);

x=(\*ptr)++;

printf("%p %d %d\n",ptr,\*ptr,x);

return 0;

}

15.

#include<stdio.h>

int main(void)

{

int x,arr[8]={11,22,33,44,55,66,77,88};

x=(arr+2)[3];

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

return 0;

}

16.

#include<stdio.h>

int main(void)

{

int arr[8]={11,22,33,44,55,66,77,88};

int \*p,\*q;

q=arr/2;

p=q\*2;

printf("%d %d",\*p,\*q);

return 0;

}

17.

#include<stdio.h>

int main(void)

{

int arr[6]={1,2,3,4,5,6};

int \*p=arr;

printf("Size of p=%u,Size of arr=%u\n",sizeof(p),sizeof(arr));

return 0;

}

18.

#include<stdio.h>

int main(void)

{

float a=5,\*p,\*\*pp;

p=&a;

pp=&p;

printf("a=%f, p=%p, pp=%p\n",a,p,pp);

a=a+1;

p=p+1;

pp=pp+1;

printf("a=%f, p=%p, pp=%p\n",a,p,pp);

return 0;

}

19.

#include<stdio.h>

int a=5,b=10;

void change1(int \*p)

{ p=&a; }

void change2(int \*\*pp)

{\*pp=&b;}

int main(void)

{

int x=20,\*ptr=&x;

printf("%d ",\*ptr);

change1(ptr);

printf("%d ",\*ptr);

change2(&ptr);

printf("%d\n",\*ptr);

return 0;

}

20.

#include<stdio.h>

void func(int x,int \*y);

int main(void)

{

int a=2,b=6;

func(a,&b);

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

return 0;

}

void func(int x,int \*y)

{

int temp;

temp=x;

x=\*y;

\*y=temp;

}

21.

#include<stdio.h>

void func(int \*\*pp);

int main(void)

{

int \*ptr;

func(&ptr);

printf("%d\n",\*ptr);

return 0;

}

void func(int \*\*pp)

{

int num=10;

\*pp=&num;

}

22.

#include<stdio.h>

void func(int x,int y);

int main(void)

{

int a=5,b=8;

func(a,b);

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

return 0;

}

void func(int x,int y)

{

int temp;

temp=\*(&x), \*(&x)=\*(&y), \*(&y)=temp;

}

23.

#include<stdio.h>

void func1(int \*ptr);

void func2(int \*\*pptr);

int main(void)

{

int arr[5]={1,2,3,4,5};

int \*p=arr;

printf("p = %p,\t",p);

func1( p);

printf("p = %p,\t",p);

func2(&p);

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

return 0;

}

void func1(int \*ptr)

{

ptr++;

}

void func2(int \*\*pptr)

{

(\*pptr)++;

}

24.

#include<stdio.h>

void func(int a[10]);

int main(void)

{

int arr[10];

func(arr);

return 0;

}

void func(int a[10])

{

int b[10],x=5;

a=&x;

b=&x;

}

25.

#include<stdio.h>

int main(void)

{

int arr[3][4];

printf("%p\t",arr);

printf("%p\t",arr[0]);

printf("%p\n",&arr[0][0]);

printf("%u\t",sizeof(arr));

printf("%u\t",sizeof(arr[0]));

printf("%u\n",sizeof(arr[0][0]));

return 0;

}

26.

#include<stdio.h>

int main(void)

{

int arr[3][4][5];

printf("%p\t",arr);

printf("%p\t",arr[0]);

printf("%p\t",arr[0][0]);

printf("%p\n",&arr[0][0][0]);

printf("%u\t",sizeof(arr));

printf("%u\t",sizeof(arr[0]));

printf("%u\t",sizeof(arr[0][0]));

printf("%u\n",sizeof(arr[0][0][0]));

return 0;

}

27.

#include<stdio.h>

void func(int a[]);

int main(void)

{

int arr[10] = {1,2,3,4,5,6,7,8,9,10};

func(arr+3);

return 0;

}

void func(int a[])

{

int i;

for(i=0; a[i]!=8; i++)

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

}

28.

#include<stdio.h>

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

int main(void)

{

int i,j;

int arr[10]={3,2,4,1,5,9,8,10,7,6};

for(i=0; i<10; i++)

for(j=0; j<10-i-1; j++)

if(\*(arr+j) > \*(arr+j+1))

swap(arr+j,arr+j+1);

for(i=0; i<10; i++)

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

printf("\n");

return 0;

}

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

{

int temp;

temp=\*b, \*b=\*c, \*c=temp;

}

29.

#include<stdio.h>

int main(void)

{

int i,arr[3][4] = {{10,11,12,13},{20,21,22,23},{30,31,32,33}};

int \*pa[3];

int (\*p)[4];

p=arr;

for(i=0; i<3; i++)

pa[i]=arr[i];

printf("%d %d %d\n",pa[0][0],pa[0][1],pa[2][3]);

printf("%d %d %d\n",p[0][0],p[0][1],p[2][3]);

return 0;

}

30.

#include<stdio.h>

#include<stdlib.h>

int \*func1(void);

int \*func2(void);

int main(void)

{

int \*ptr1,\*ptr2;

ptr1=func1();

ptr2=func2();

printf("%d %d\n",\*ptr1,\*ptr2);

free(ptr2);

return 0;

}

int \*func1(void)

{

int a=8,\*p=&a;

return p;

}

int \*func2(void)

{

int \*p;

p=(int \*)malloc(sizeof(int));

\*p=9;

return p;

}

31.

#include<stdio.h>

int main(void)

{

int i,arr[3][4]={ {10,11,12,13},{20,21,22,23},{30,31,32,33}};

int \*p=&arr[0][0];

for(i=0; i<12; i++)

printf("%d ",p[i]);

printf("\n");

return 0;

}

32.

#include<stdio.h>

int main(void)

{

int a[2][3];

a[1][2]=9;

printf("%d\n",a[1,2]);

return 0;

}

33.

#include<stdio.h>

int main(void)

{

int a[5]={1},b[5]={1};

if(a==b)

printf("Same\n");

else

printf("Different\n");

return 0;

}

Exercise- Strings 10

1.

#include<stdio.h>

int main(void)

{

int i=0;

char name[10] = {'M','o','h','i','n','i','\0'};

while(name[i])

putchar(name[i++]);

return 0;

}

2.

#include<stdio.h>

int main(void)

{

char \*str;

printf("Enter a string : ");

gets(str);

printf("String is %s\n",str);

return 0;

}

3.

#include<stdio.h>

#include<string.h>

int main(void)

{

char \*str1="Good",\*str2="Morning";

strcat(str1,str2);

printf("%s\n",str1);

return 0;

}

4.

#include<stdio.h>

#include<string.h>

int main(void)

{

char str[10]="How";

strcat(str,'?');

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

return 0;

}

5.

#include<stdio.h>

int main(void)

{

char str[]="Vijaynagar";

str=str+5;

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

return 0;

}

6.

#include<stdio.h>

void func(char str[]);

int main(void)

{

char str[]="Vijaynagar";

func(str);

return 0;

}

void func(char str[])

{

str=str+5;

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

}

7.

#include<stdio.h>

int main(void)

{

char str[] ={70,97,105,116,104,0};

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

return 0;

}

8.

#include<stdio.h>

int main(void)

{

char \*p="Devanshi";

char arr[]="Devanshi";

\*(p+2)='b';

arr[2]='b';

puts(p);

puts(arr);

return 0;

}

9.

#include<stdio.h>

int main(void)

{

char str[]="painstaking";

char \*p=str+5;

printf("%c\t",\*p);

printf("%s\n",p);

return 0;

}

10.

#include<stdio.h>

int main(void)

{

printf("%c\t","Determination"[2]);

printf("%c\t",\*("Determination"+2));

printf("%s\t","Determination"+2);

printf("Determination"+2);

printf("\t");

printf("Determination"+strlen("Deepali"));

printf("\t");

printf("Determination"+sizeof("Deepali"));

printf("\n");

return 0;

}

11.

#include<stdio.h>

int main(void)

{

char str[]="Lucknow";

char \*p=str;

p=p+3;

p[3]='t';

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

return 0;

}

12.

#include<stdio.h>

#include<string.h>

int main(void)

{

char \*p[]={"Orange","Yellow","Sky""Blue","Black"};

char arr[10];

printf("%s %s %s\n",p[1],p[2],p[3]);

strcpy(arr,"Luck""now");

printf("%s\n",arr);

return 0;

}

13.

#include<stdio.h>

#include<string.h>

int main(void)

{

char str1[15]="Good ";

char str2[]="Evening";

strcpy(str1+strlen(str1),str2);

printf("%s\n",str1);

return 0;

}

14.

#include<stdio.h>

#include<string.h>

int main(void)

{

char str1[]="Parul";

char str2[10];

strcpy(str2,str1);

if(str1==str2)

printf("Same\n");

else

printf("Different\n");

return 0;

}

15.

#include<stdio.h>

int main(void)

{

char x[]="Shilpee";

char y[20];

y="Anjali";

printf("%s %s\n",x,y);

return 0;

}

16.

#include<stdio.h>

#include<string.h>

int main(void)

{

char str1[]="deep";

char str2[]={'d','e','e','p'};

if(strcmp(str1,str2)==0)

printf("Same\n");

else

printf("Different\n");

return 0;

}

17.

#include<stdio.h>

#include<string.h>

int main(void)

{

char str1[]="Parul",str2[]="Devanshi";

if(strlen(str1)-strlen(str2) >=0 )

puts(str1);

else

puts(str2);

return 0;

}

18.

#include<stdio.h>

#include<string.h>

int main(void)

{

char name[15]="Vikramaditya";

int i=0;

while(name[i])

{

printf("%c ",name[i]);

i=i+3;

}

return 0;

}

19.

#include<stdio.h>

int main(void)

{

char str[10][20];

int i;

for(i=0; i<10; i++)

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

for(i=0; i<10; i++)

printf("%s ",str[i]);

return 0;

}

20

#include<stdio.h>

int main(void)

{

char \*str[10];

int i;

for(i=0; i<10; i++)

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

for(i=0; i<10; i++)

printf("%s",str[i]);

return 0;

}

21.

#include<stdio.h>

#include<string.h>

char \*combine(char \*arr1,char \*arr2);

int main(void)

{

char str1[20],str2[20];

char \*p;

strcpy(str1,"Suresh ");

strcpy(str2,"Kumar");

p=combine(str1,str2);

puts(p);

return 0;

}

char \*combine(char \*arr1,char \*arr2)

{

char str[80];

strcpy(str,arr1);

strcat(str,arr2);

return str;

}

22.

#include<stdio.h>

int main(void)

{

char \*str="Deepali Srivastava";

int i=0;

while(str[++i]);

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

return 0;

}

23.

#include<stdio.h>

int main(void)

{

int d1,m1,y1;

char date[11]="24/05/1973";

date[2]=date[5]='\0';

sscanf(date,"%d",&d1);

sscanf(date+3,"%d",&m1);

sscanf(date+6,"%d",&y1);

date[2]=date[5]='/';

printf("d1=%d,m1=%d,y1=%d\n",d1,m1,y1);

printf("date=%s\n",date);

return 0;

}

24.

#include<stdio.h>

void func(char \*p);

int main(void)

{

char \*str="doubtful";

func(str);

return 0;

}

void func(char \*p)

{

if(\*p!='f')

{

printf("%c",\*p);

func(++p);

}

}

25.

#include<stdio.h>

void func(char \*p);

int main(void)

{

char \*str="tap";

func(str);

return 0;

}

void func(char \*p)

{

if(\*p)

{

func(p+1);

printf("%c",\*p++);

}

}

26.

#include<stdio.h>

int main(void)

{

char \*ptr;

ptr="My name is %s and age is %d\n";

printf(ptr,"Ranju",30);

return 0;

}

27.

#include<stdio.h>

void func1(char x[]);

void func2(char x[]);

int main(void)

{

char arr[5];

puts(arr);

func1(arr);

puts(arr);

func2(arr);

puts(arr);

return 0;

}

void func1(char x[])

{

x="Jack";

puts(x);

}

void func2(char x[])

{

x[0]='J', x[1]='i', x[2]='l', x[3]='l', x[4]='\0';

puts(x);

}

28.

#include<stdio.h>

int main(void)

{

char \*ptr;

ptr = "Every saint has a past,\

Every sinner has a future.\n";

printf("Giving " "is " "living.""\n");

printf(ptr);

return 0;

}

29.

#include<stdio.h>

int main(void)

{

int marks;

char name[50];

printf("Enter marks : ");

scanf("%d",&marks);

printf("Enter name : ");

gets(name);

puts(name);

return 0;

}

30.

#include<stdio.h>

#include<string.h>

int main(void)

{

char \*p, str[100]=" main()";

strcpy(str, str+strspn(str, " \t"));

puts(str);

return 0;

}

31.

#include<stdio.h>

#include<ctype.h>

char \*str\_upper(char \*str);

int main(void)

{

char str[50];

printf("Enter a string : ");

gets(str);

puts(str\_upper(str));

return 0;

}

char \*str\_upper(char \*str)

{

char \*s=str;

while(\*s!='\0')

{

\*s=toupper(\*s);

s++;

}

return str;

}

32.

#include<stdio.h>

#include<ctype.h>

int strCIcmp(char \*str1,char \*str2);

int main(void)

{

char str1[50]="Deepali",str2[20]="deePali";

printf("%d\n",strCIcmp(str1,str2));

return 0;

}

int strCIcmp(char \*str1,char \*str2)

{

while(toupper(\*str1) == toupper(\*str2))

{

if(\*str1 == '\0')

return 0;

str1++;

str2++;

}

return(toupper(\*str1) - toupper(\*str2));

}

33.

#include<stdio.h>

#include<string.h>

int main(void)

{

char \*p,name[50];

printf("Enter a string : ");

fgets(name,sizeof(name),stdin);

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

printf("Enter a string : ");

fgets(name,sizeof(name),stdin);

if((p=strchr(name,'\n'))!=NULL)

\*p='\0';

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

printf("Enter a string : ");

fgets(name,sizeof(name),stdin);

name[strlen(name)-1]='\0';

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

return 0;

}

34.

#include<stdio.h>

#include<string.h>

char \*remove\_LTblanks(char \*str);

int main(void)

{

char str[100]=" Deepali Saxena ";

printf("...%s..",remove\_LTblanks(str));

return 0;

}

char \*remove\_LTblanks(char \*str)

{

int i,j,l,t,newlength;

l=0;

while(str[l]==' ')

l++;

/\*Leading Blanks = l\*/

j=strlen(str)-1;

while(str[j]==' ')

j--;

t=strlen(str)-1-j;

/\*printf("Trailing Blanks = t\*/

newlength = strlen(str)-l-t;

for(i=0,j=l; i<newlength; i++,j++)

str[i]=str[j];

str[newlength]='\0';

return str;

}

35.

#include<stdio.h>

#include<string.h>

#include<ctype.h>

char \*sort(char \*str);

int main(void)

{

char str[30];

strcpy(str,"Devanshi");

puts(sort(str));

return 0;

}

char \*sort(char \*str)

{

int i,j,n;

char temp;

n=strlen(str);

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

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

{

if(toupper(str[j]) > toupper(str[j+1]) )

{

temp = str[j];

str[j] = str[j+1];

str[j+1] = temp;

}

}

return str;

}

36.

#include<stdio.h>

#include<string.h>

#include<ctype.h>

void abbreviate(char \*str1,char \*str2);

int main(void)

{

char str1[50],str2[10];

strcpy(str1," World Health Organisation ");

abbreviate(str1,str2);

puts(str2);

return 0;

}

void abbreviate(char \*str1,char \*str2)

{

while(\*str1!='\0')

{

while(isspace(\*str1)) /\*Skip white spaces\*/

str1++;

if(\*str1=='\0')

{

\*str2='\0';

return;

}

\*str2++ = \*str1;

while(!isspace(\*str1) && \*str1!='\0')

str1++;

}

\*str2='\0';

}

37.

#include<stdio.h>

#include<string.h>

void extract(char \*str1,int i,int n,char \*str2);

int main(void)

{

char str1[50],str2[50];

strcpy(str1,"Srivastava");

extract(str1,3,6,str2);

puts(str2);

return 0;

}

void extract(char \*str1,int i, int n, char \*str2)

{

int j,k;

j=0;

while(str1[j]!='\0' && j<i)

j++;

for(k=0; k<n; k++,j++)

str2[k]=str1[j];

str2[k]='\0';

}

38.

#include<stdio.h>

#include<string.h>

void del\_Multspaces(char \*str);

int main(void)

{

char str[100]=" Data Structures through C in depth ";

del\_Multspaces(str);

puts(str);

return 0;

}

void del\_Multspaces(char \*str)

{

int i=0,j=0;

while(i<=strlen(str)-1)

{

if(str[i]==' ')

{

while(str[i]==' ')

i++;

str[j++]=' ';

}

str[j++]=str[i++];

}

str[j]='\0';

}

39.

#include<stdio.h>

#include<string.h>

int pstrncmp(char \*str1,char \*str2, int n);

char \*pstrncpy(char \*str1, char \*str2, int n);

char \*pstrncat(char \*str1,char \*str2, int n);

int main(void)

{

char str1[50]="Dev", str2[50]="anshikal";

pstrncat(str1,str2,5);

puts(str1);

strcpy(str1,"Deepali"); strcpy(str2,"Deepam");

printf("%d\n",pstrncmp(str1,str2,5) );

strcpy(str1,""); strcpy(str2,"Vinay");

pstrncpy(str1,str2,4);

puts(str1);

return 0;

}

int pstrncmp(char \*str1,char \*str2, int n)

{

while(\*str1 == \*str2)

{

n--;

if(\*str1 == '\0' || n<=0 )

return 0;

str1++;

str2++;

}

return(\*str1 - \*str2);

}

char \*pstrncpy(char \*str1, char \*str2, int n)

{

while(\*str2!='\0' && n>0)

{

\*str1++ = \*str2++;

n--;

}

while(n > 0);

{

\*str1++ = '\0';

n--;

}

return str1;

}

char \*pstrncat(char \*str1,char \*str2, int n)

{

char \*p=str1;

int i;

while(\*p!='\0')

p++;

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

\*p++ = \*str2++;

\*str2='\0';

return str1;

}

40.

#include<stdio.h>

#include<string.h>

char \*strstr\_r(char \*str, char \*substr);

int main(void)

{

char str1[50];

strcpy(str1,"no yes ... yes no yes no");

puts(strstr\_r(str1,"yes"));

return 0;

}

/\*Return a pointer to the last occurrence of the substring\*/

char \*strstr\_r(char \*str, char \*substr)

{

char \*prev,\*s;

if(\*substr=='\0')

return NULL;

prev=NULL;

s=strstr(str,substr);

while(s!=NULL)

{

prev=s;

s=strstr(prev+1,substr);

}

return prev;

}

41.

#include<stdio.h>

#include<string.h>

int find\_indexF(char \*str, char \*substr);

int find\_indexL(char \*str, char \*substr);

int main(void)

{

char str1[30];

strcpy(str1,"no yes no yes");

printf("%d\n",find\_indexF(str1,"yes"));

printf("%d\n",find\_indexL(str1,"yes"));

return 0;

}

/\*Returns index of the first occurrence of substring\*/

int find\_indexF(char \*str, char \*substr)

{

char \*s,\*p1,\*p2;

if(\*substr=='\0') /\*If substring is empty\*/

return -1;

for(s=str; \*s!='\0'; s++)

{

p1=s;

p2=substr;

while(\*p2!='\0' && \*p1==\*p2)

{

p1++;

p2++;

}

if(\*p2=='\0')

return s-str;

}

return -1;

}

/\*Returns index of the last occurrence of substring\*/

int find\_indexL(char \*str, char \*substr)

{

int i;

char \*s,\*p1,\*p2;

if(\*substr=='\0') /\*If substring is empty\*/

return -1;

i=-1;

for(s=str; \*s!='\0'; s++)

{

p1=s;

p2=substr;

while(\*p2!='\0' && \*p1==\*p2)

p1++,p2++;

if(\*p2=='\0')

i = s-str;

}

return i;

}

42.

#include<stdio.h>

#include<string.h>

int str\_start(char \*str1, char \*str2);

int str\_end(char \*str, char \*substr);

int main(void)

{

char str1[30]="yes no yes";

int i;

i=str\_start(str1,"yes");

printf("%d ",i);

i=str\_start(str1,"es");

printf("%d ",i);

i=str\_end(str1,"yes");

printf("%d ",i);

i=str\_end(str1,"ye");

printf("%d ",i);

return 0;

}

/\*Returns 1 if substr is present at the start of str\*/

int str\_start(char \*str, char \*substr)

{

char \*p1,\*p2;

if(\*substr=='\0')

return 0;

p1=str;

p2=substr;

while(\*p1 == \*p2)

{

if(\*p1=='\0' || \*p2=='\0')

break;

p1++;

p2++;

}

if(\*p2=='\0')

return 1;

else

return 0;

}

/\*Returns 1 if substr is present at the end of str\*/

int str\_end(char \*str, char \*substr)

{

char \*p1,\*p2;

if(\*substr=='\0')

return 0;

for(p1=str; \*p1!='\0'; p1++)

;

for(p2=substr; \*p2!='\0'; p2++)

;

/\*Now p1 and p2 point to the end of str and substr respectively\*/

while(\*p1 == \*p2)

{

if(p1==str || p2==substr)

break;

p1--;

p2--;

}

if(\*p1==\*p2 && p2==substr)

return 1;

else

return 0;

}

43.

#include<stdio.h>

#include<string.h>

int main(void)

{

char \*p,x,str[100]="JP Nagar Bangalore, Vijaynagar Bangalore, Jaynagar Bangalore";

p=strstr(str,"Bangalore");

while(p!=NULL)

{

x=p[9];

strcpy(p,"Bengaluru");

p[9]=x; /\*we dont need the '\0' character written by strcpy()\*/

p=strstr(str,"Bangalore");

}

puts(str);

return 0;

}

44.

#include<stdio.h>

#include<string.h>

int count\_word1(char \*str,char \*word);

int count\_word2(char \*str,char \*word);

int main(void)

{

char str[100]="hut cut hut but nuthut hut on hutin but hut";

printf("\n%d \n",count\_word1(str,"hut"));

printf("\n%d \n",count\_word2(str,"hut"));

return 0;

}

int count\_word1(char \*str,char \*word)

{

int i=0;

char \*s=str;

s=strstr(s,word);

while(s!=NULL)

{

if(s==str && \*(s+strlen(word))=='\0') /\*only word\*/

i++;

else if(s==str && \*(s+strlen(word))==' ')/\*first word\*/

i++;

else if(\*(s-1)==' ' && \*(s+strlen(word))=='\0' )/\*last word\*/

i++;

else if(\*(s-1)==' ' && \*(s+strlen(word))==' ')/\*in middle\*/

i++;

if(s)

puts(s);

s=strstr(s+1,word);

}

return i;

}

int count\_word2(char \*str,char \*word)

{

int i=0;

char \*s=str;

s=strstr(s,word);

while(s!=NULL)

{

if(s==str || \*(s-1)==' ')

if(\*(s+strlen(word))=='\0'|| \*(s+strlen(word))==' ')

i++;

if(s)

puts(s);

s=strstr(s+1,word);

}

return i;

}

45.

#include<stdio.h>

#include<string.h>

void func(char \*str1, char \*str2);

int main(void)

{

char str1[100],str2[100];

strcpy(str1,"Deepali Sri");strcpy(str2,"Suresh Sri");

func(str1,str2);

puts(str1);

puts(str2);

return 0;

}

void func(char \*str1, char \*str2)

{

char str3[80],str4[80];

int i,j,k,len1,len2;

len1=strlen(str1);

len2=strlen(str2);

k=0;

for(i=0; i<len1; i++)

{

for(j=0; j<len2; j++)

if(str1[i] == str2[j])

break;

if(j==len2)

str3[k++] = str1[i];

}

str3[k]='\0';

k=0;

for(i=0; i<len2; i++)

{

for(j=0; j<len1; j++)

if(str2[i]==str1[j])

break;

if(j==len1)

str4[k++]=str2[i];

}

str4[k] = '\0';

strcpy(str1,str3);

strcpy(str2,str4);

}

46.

#include<stdio.h>

int main(void)

{

int d,m,y,j,f,h,fh,day;

char \*days[] = {"Saturday","Sunday", "Monday", "Tuesday", "Wednesday", "Thursday","Friday"};

printf("Enter date(dd/mm/yyyy): ");

scanf("%d/%d/%d",&d,&m,&y);

j = d;

switch(m-1)

{

case 11: j+=30;

case 10: j+=31;

case 9: j+=30;

case 8: j+=31;

case 7: j+=31;

case 6: j+=30;

case 5: j+=31;

case 4: j+=30;

case 3: j+=31;

case 2: j+=28;

case 1: j+=31;

}

if(y%4==0 && y%100!=0 || y%400==0)

if(m!=1 && m!=2)

j = j+1;

f = (y-1)/4;

h = (y-1)/100;

fh = (y-1)/400;

day = (y+j+f-h+fh)%7;

printf("%s\n",days[day]);

return 0;

}/\*End of main()\*/

47.

#include<stdio.h>

#include<string.h>

int count\_vowels (char \*str);

int main(void)

{

char str[100];

printf("Enter a string :");

gets(str);

printf("%d\n",countVowels(str));

return 0;

}

int countVowels(char \*str)

{

if(\*str == '\0')

return 0;

switch(\*str)

{

case 'A': case 'a':

case 'E': case 'e':

case 'I': case 'i':

case 'O': case 'o':

case 'U': case 'u':

return 1 + countVowels(str+1);

default:

return countVowels(str+1);

}

}

48.

#include<stdio.h>

void f(char \*s, char a, char b);

int main(void)

{

char str[100],a,b;

printf("Enter a string :");

gets(str);

printf("Enter two characters :");

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

f(str,a,b);

puts(str);

return 0;

}

void f(char \*str, char a, char b)

{

if(\*str=='\0')

return;

if(\*str==a)

\*str=b;

f(str+1,a,b);

}

49.

#include<stdio.h>

#include<string.h>

void reverse\_str(char \*str);

void rev(char \*s,int size);

int main(void)

{

char str[100];

printf("Enter a string :");

gets(str);

reverse\_str(str);

puts(str);

return 0;

}

void reverse\_str(char \*str)

{

rev(str,strlen(str));

}

void rev(char \*str,int size)

{

char tmp;

if(size<=1)

return;

else

{

tmp = str[0];

str[0] = str[size-1];

str[size-1] = tmp;

rev(str+1,size-2);

}

}

50.

#include<stdio.h>

#include<string.h>

int find(char \*str, char c, int i);

int findFirst(char \*str, char c);

int main(void)

{

char str[100];

printf("Enter a string :");

gets(str);

printf("%d\n",findFirst(str,'a'));

return 0;

}

int findFirst(char \*str, char c)

{

return find(str,c,0);

}

int find(char \*str, char c, int i)

{

if(\*str=='\0')

return -1;

if(\*str == c)

return i;

return find(str+1,c,i+1);

}

51.

#include<stdio.h>

#include<string.h>

int findLast(char \*str, char c);

int find(char \*str, char c, int n);

int main(void)

{

char str[100];

char c='a';

printf("Enter a string :");

gets(str);

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

return 0;

}

int findLast(char \*str,char c)

{

return find(str,c,strlen(str));

}

int find(char \*str, char c, int n)

{

if(n==0)

return -1;

if(str[n-1]==c)

return n-1;

return find(str,c,n-1);

}

52.

#include<stdio.h>

#include<string.h>

int is\_palindrome(char \*s);

int find(char \*s, int size);

int main(void)

{

char str[100];

printf("Enter a string :");

gets(str);

if(is\_palindrome(str))

printf("%s is a palindrome\n", str);

else

printf("%s is not a palindrome\n", str);

return 0;

}

int is\_palindrome(char \*str)

{

return find(str,strlen(str));

}

int find(char \*str, int size)

{

if(size<=1)

return 1;

if(str[0] != str[size-1])

return 0;

return find(str+1,size-2);

}

53.

#include<stdio.h>

#include<string.h>

#include<ctype.h>

int is\_palindrome(char \*s);

int find(char \*s, int size);

int main(void)

{

char str[100];

printf("Enter a string :");

gets(str);

if(is\_palindrome(str))

printf("%s is a palindrome\n", str);

else

printf("%s is not a palindrome\n", str);

puts(str);

return 0;

}

int is\_palindrome(char \*str)

{

return find(str,strlen(str));

}

int find(char \*str, int size)

{

if(size<=1)

return 1;

if(str[0] == ' ' || str[0]==',' || str[0]=='"'|| str[0]=='.'|| str[0]=='!'|| str[0]=='-')

return find(str+1, size-1);

if(str[size-1]==' ' || str[size-1]==','|| str[size-1]=='"'|| str[size-1]=='.'|| str[size-1]=='!'|| str[size-1]=='-')

return find(str, size-1);

if(toupper(str[0]) != toupper(str[size-1]) )

return 0;

return find(str+1, size-2);

}

54.

#include<stdio.h>

#include<ctype.h>

int str\_to\_i(char \*str);

void f(char \*s, int \*num);

int main(void)

{

char str[10];

int num;

printf("Enter a string of numbers :");

gets(str);

num=str\_to\_i(str);

printf("%d %s\n",num,str);

return 0;

}

int str\_to\_i(char \*str)

{

int num=0;

f(str,&num);

return num;

}

void f(char \*s, int \*pnum)

{

if(\*s=='\0' || !isdigit(\*s) )

return;

\*pnum = (\*pnum)\*10 + \*s-'0';

return f(s+1, pnum);

}

55.

#include<stdio.h>

#include<string.h>

void Permute1(char str[]);

void Permute1\_rec(char str[], char\* currentptr);

void Permute2(char str[]);

void Permute2\_rec(char str[], int startIndex, int lastIndex);

void Swap(char \*a, char \*b);

int main(void)

{

char str[10]="abc";

Permute1(str);

printf("\n\n");

Permute2(str);

printf("\n");

return 0;

}

void Permute1(char str[])

{

Permute1\_rec(str,str);

}

void Permute1\_rec(char str[], char\* currentptr)

{

char \*ptr;

if( \*(currentptr + 1) == '\0')

printf("%s\t", str);

else

for(ptr=currentptr; \*ptr!='\0'; ptr++)

{

Swap(ptr,currentptr);

Permute1\_rec(str, currentptr+1);

Swap(ptr,currentptr);

}

}

void Permute2(char str[])

{

Permute2\_rec(str,0,strlen(str)-1);

}

void Permute2\_rec(char str[], int startIndex, int lastIndex)

{

int i;

if(startIndex==lastIndex)

{

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

printf("%c",str[i]);

printf("\t");

}

else

for(i=startIndex;i<=lastIndex;i++)

{

Swap(&str[startIndex], &str[i]);

Permute2\_rec(str,startIndex+1,lastIndex);

Swap(&str[startIndex], &str[i]);

}

}

void Swap(char \*a, char \*b)

{

char temp = \*a; \*a=\*b; \*b=temp;

}

Exercise- Structurs and Unions

1.

#include<stdio.h>

int main(void)

{

struct result

{

int marks;

char grade;

};

struct result A1, B1;

A1.marks=80; A1.grade='A';

B1=A1;

printf("Marks=%d\t",B1.marks);

printf("Grade=%c\n",B1.grade);

return 0;

}

2.

#include<stdio.h>

#include<stdlib.h>

int main(void)

{

struct rec

{

char \*name;

int age;

}\*ptr;

char name[10]="Somalika";

ptr=(struct rec \*)malloc(sizeof(struct rec));

ptr->name=name;

ptr->age=93;

printf("%s\t",ptr->name);

printf("%d\n",ptr->age);

return 0;

}

3.

#include<stdio.h>

struct student {char name[20]; int age;};

int main(void)

{

struct student stu1={"Anita", 10},stu2={"Anita",12};

if(stu1 == stu2)

printf("Same\n");

else

printf("Not same\n");

return 0;

}

4.

#include<stdio.h>

void func(struct tag v);

int main(void)

{

struct tag

{

int i;

char c;

};

struct tag var={2,'s'};

func(var);

return 0;

}

void func(struct tag v)

{

printf("%d %c\n",v.i,v.c);

}

5.

#include<stdio.h>

void func(struct {int i; char c;} v);

int main(void)

{

struct {int i; char c;}var = {2,'s'};

func(var);

return 0;

}

void func(struct {int i; char c;} v)

{

printf("%d %c\n",v.i,v.c);

}

6.

#include<stdio.h>

struct tag{int i; char c;};

void func(struct tag);

int main(void)

{

struct tag var = {12,'c'};

func(var);

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

return 0;

}

void func(struct tag var)

{

var.i++;

}

7.

#include<stdio.h>

struct tag{ int i; char c;};

void func(struct tag \*);

int main(void)

{

struct tag var = {12,'c'};

func(&var);

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

return 0;

}

void func(struct tag \*ptr)

{

ptr->i++;

}

8.

#include<stdio.h>

#include<string.h>

int main(void)

{

union tag

{

char name[15];

int age;

}rec;

strcpy(rec.name,"Somalika");

rec.age=23;

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

return 0;

}

9.

#include<stdio.h>

struct

{

char a[20];

int b;

union

{

double c;

struct

{

char d[15];

float e;

}x;

}y;

}z;

int main(void)

{

printf("%u %u %u\n",sizeof(z.y.x),sizeof(z.y),sizeof(z));

return 0;

}

10.

#include<stdio.h>

int main(void)

{

typedef short int s\_int;

unsigned s\_int var=3;

printf("%u", var);

return 0;

}

11.

#include<stdio.h>

typedef struct tag{int i; char c;}tag;

int main(void)

{

struct tag v1={1,'a'};

tag v2={2,'b'};

printf("%d %c %d %c\n",v1.i,v1.c,v2.i,v2.c);

return 0;

}

12.

#include<stdio.h>

typedef struct{char name[20]; int age;}stu;

typedef struct{int data; node \*link;}node;

int main(void)

{

stu \*p=malloc(sizeof(stu));

node \*ptr=malloc(sizeof(node));

p->age=30;

ptr->data=3;

printf("%d %d\n",p–>age,ptr–>data);

return 0;

}

13.

#include<stdio.h>

#include<limits.h>

#define N 5

struct person

{

char name[20];

int age;

char city[50];

};

int main(void)

{

struct person p[5],eldestP;

int i,max=INT\_MIN;

for(i=0; i<5; i++)

{

printf("Enter name : ");

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

printf("Enter age : ");

scanf("%d",&p[i].age);

printf("Enter city : ");

scanf("%s",p[i].city);

if(p[i].age > max)

{

max=p[i].age;

eldestP=p[i];

}

}

printf("%s %d %s\n",eldestP.name,eldestP.age,eldestP.city);

return 0;

}

14.

#include<stdio.h>

#define N 5

struct train

{

char name[20];

int hr;

int min;

char m;

};

int main(void)

{

struct train t[5];

int i;

for(i=0; i<5; i++)

{

printf("Enter name : ");

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

printf("Enter arrival time(hh:mm A/P) : ");

scanf("%d : %d %c",&t[i].hr, &t[i].min,&t[i].m);

}

for(i=0; i<5; i++)

{

printf("%s\t",t[i].name);

t[i].m=='A' ? printf("%d",t[i].hr) : printf("%d",t[i].hr+12);

printf(":%d\n",t[i].min);

}

return 0;

}

15.

#include<stdio.h>

#include<string.h>

#define N 5

struct employee

{

char name[20];

int age;

int salary;

};

void sort(struct employee emp[]);

void display(struct employee emp[]);

int main(void)

{

struct employee emp[N];

int i;

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

{

printf("Enter name :");

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

printf("Enter age : ");

scanf("%d", &emp[i].age);

printf("Enter salary : ");

scanf("%d", &emp[i].salary);

printf("\n");

}

display(emp);

sort(emp);

display(emp);

return 0;

}

void sort(struct employee emp[])

{

struct employee temp;

int i,j;

for(i=0; i<N-1; i++)

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

if(strcmp(emp[i].name, emp[j].name) > 0 )

{

temp = emp[i];

emp[i] = emp[j];

emp[j] = temp;

}

}

void display(struct employee emp[])

{

int i;

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

{

printf("%s\t\t",emp[i].name);

printf("%d\t", emp[i].age);

printf("%d\n", emp[i].salary);

}

printf("\n");

}

16.

#include<stdio.h>

#include<stdlib.h>

struct node

{

int info;

struct node \*link;

};

struct node \*create\_list(struct node \*start);

void display(struct node \*start);

struct node \*addatbeg(struct node \*start,int data);

struct node \*addatend(struct node \*start,int data);

int countOccurrences(struct node \*start, int n);

int main(void)

{

struct node \*start=NULL;

int n;

start=create\_list(start);

display(start);

printf("Enter a value : ");

scanf("%d",&n);

printf("The value %d occurs %d times\n",n,countOccurrences(start,n) );

return 0;

}/\*End of main()\*/

int countOccurrences(struct node \*ptr, int n)

{

int k=0;

while(ptr!=NULL)

{

if(ptr->info == n)

k++;

ptr=ptr->link;

}

return k;

}/\*End of countOccurrences()\*/

struct node \*create\_list(struct node \*start)

{

int i,n,data;

printf("Enter the number of nodes : ");

scanf("%d",&n);

start=NULL;

if(n==0)

return start;

printf("Enter the element to be inserted : ");

scanf("%d",&data);

start=addatbeg(start,data);

for(i=2;i<=n;i++)

{

printf("Enter the element to be inserted : ");

scanf("%d",&data);

start=addatend(start,data);

}

return start;

}/\*End of create\_list()\*/

void display(struct node \*start)

{

struct node \*p;

if(start==NULL)

{

printf("List is empty\n");

return;

}

p=start;

printf("List is :\n");

while(p!=NULL)

{

printf("%d ",p->info);

p=p->link;

}

printf("\n\n");

}/\*End of display() \*/

struct node \*addatbeg(struct node \*start,int data)

{

struct node \*tmp;

tmp=(struct node \*)malloc(sizeof(struct node));

tmp->info=data;

tmp->link=start;

start=tmp;

return start;

}/\*End of addatbeg()\*/

struct node \*addatend(struct node \*start,int data)

{

struct node \*p,\*tmp;

tmp=(struct node \*)malloc(sizeof(struct node));

tmp->info=data;

p=start;

while(p->link!=NULL)

p=p->link;

p->link=tmp;

tmp->link=NULL;

return start;

}/\*End of addatend()\*/

17.

#include<stdio.h>

#include<stdlib.h>

struct node

{

int info;

struct node \*link;

};

struct node \*create\_list(struct node \*start);

void display(struct node \*start);

struct node \*addatbeg(struct node \*start,int data);

struct node \*addatend(struct node \*start,int data);

int largest(struct node \*start);

int smallest(struct node \*start);

int main(void)

{

struct node \*start=NULL;

start=create\_list(start);

display(start);

printf("Largest element is %d\n",largest(start));

printf("Smallest element is %d\n",smallest(start));

return 0;

}/\*End of main()\*/

int largest(struct node \*ptr)

{

int large=ptr->info;

while(ptr!=NULL)

{

if(ptr->info >large)

large = ptr->info;

ptr=ptr->link;

}

return large;

}/\*End of largest()\*/

int smallest(struct node \*ptr)

{

int small=ptr->info;

while(ptr!=NULL)

{

if(ptr->info < small)

small = ptr->info;

ptr=ptr->link;

}

return small;

}/\*End of smallest()\*/

struct node \*create\_list(struct node \*start)

{

int i,n,data;

printf("Enter the number of nodes : ");

scanf("%d",&n);

start=NULL;

if(n==0)

return start;

printf("Enter the element to be inserted : ");

scanf("%d",&data);

start=addatbeg(start,data);

for(i=2;i<=n;i++)

{

printf("Enter the element to be inserted : ");

scanf("%d",&data);

start=addatend(start,data);

}

return start;

}/\*End of create\_list()\*/

18.

#include<stdio.h>

#include<stdlib.h>

struct node

{

int info;

struct node \*link;

};

struct node \*create\_list(struct node \*start);

void display(struct node \*start);

struct node \*addatbeg(struct node \*start,int data);

struct node \*addatend(struct node \*start,int data);

struct node \*Copy(struct node \*start1);

int main(void)

{

struct node \*start1=NULL,\*start2;

printf("Enter list 1 -\n");

start1=create\_list(start1);

printf("List 1 is :\n");

display(start1);

start2 = Copy(start1);

printf("Copy of list 1 is :\n");

display(start2);

return 0;

}/\*End of main()\*/

struct node \*Copy(struct node \*start)

{

struct node \*startCopy=NULL;

struct node \*p1,\*p2,\*tmp;

if(start==NULL)

return NULL;

p1=start;

tmp=(struct node \*)malloc(sizeof(struct node));

tmp->info=p1->info;

p2=startCopy=tmp;

p1=p1->link;

while(p1!=NULL)

{

tmp=(struct node \*)malloc(sizeof(struct node));

tmp->info = p1->info;

p2->link=tmp;

p2=tmp;

p1=p1->link;

}

p2->link=NULL;

return startCopy;

}/\*End of Copy()\*/

struct node \*create\_list(struct node \*start)

{

int i,n,data;

printf("Enter the number of nodes : ");

scanf("%d",&n);

start=NULL;

if(n==0)

return start;

printf("Enter the element to be inserted : ");

scanf("%d",&data);

start=addatbeg(start,data);

for(i=2;i<=n;i++)

{

printf("Enter the element to be inserted : ");

scanf("%d",&data);

start=addatend(start,data);

}

return start;

}/\*End of create\_list()\*/

void display(struct node \*start)

{

struct node \*p;

if(start==NULL)

{

printf("Empty\n");

return;

}

p=start;

while(p!=NULL)

{

printf("%d ",p->info);

p=p->link;

}

printf("\n\n");

}/\*End of display() \*/

struct node \*addatbeg(struct node \*start,int data)

{

struct node \*tmp;

tmp=(struct node \*)malloc(sizeof(struct node));

tmp->info=data;

tmp->link=start;

start=tmp;

return start;

}/\*End of addatbeg()\*/

struct node \*addatend(struct node \*start,int data)

{

struct node \*p,\*tmp;

tmp=(struct node \*)malloc(sizeof(struct node));

tmp->info=data;

p=start;

while(p->link!=NULL)

p=p->link;

p->link=tmp;

tmp->link=NULL;

return start;

}/\*End of addatend()\*/

19.

#include<stdio.h>

#include<stdlib.h>

struct node

{

int info;

struct node \*link;

};

struct node \*create\_list(struct node \*start);

void display(struct node \*start);

struct node \*addatbeg(struct node \*start,int data);

struct node \*addatend(struct node \*start,int data);

void MoveLarge(struct node \*start);

int main(void)

{

struct node \*start=NULL;

start=create\_list(start);

display(start);

MoveLarge(start);

display(start);

return 0;

}/\*End of main()\*/

void MoveLarge(struct node \*start)

{

struct node \*p;

int tmp;

if(start==NULL)

return;

p=start;

while(p->link!=NULL)

{

if(p->info > p->link->info)

{

tmp = p->info;

p->info = p->link->info;

p->link->info = tmp;

}

p=p->link;

}

}/\*End of MoveLarge()\*/

struct node \*create\_list(struct node \*start)

{

int i,n,data;

printf("Enter the number of nodes : ");

scanf("%d",&n);

start=NULL;

if(n==0)

return start;

printf("Enter the element to be inserted : ");

scanf("%d",&data);

start=addatbeg(start,data);

for(i=2;i<=n;i++)

{

printf("Enter the element to be inserted : ");

scanf("%d",&data);

start=addatend(start,data);

}

return start;

}/\*End of create\_list()\*/

20. #include<stdio.h>

#include<stdlib.h>

struct node

{

int info;

struct node \*link;

};

struct node \*create\_list(struct node \*start);

void display(struct node \*start);

struct node \*addatbeg(struct node \*start,int data);

struct node \*addatend(struct node \*start,int data);

void MoveSmall(struct node \*start);

int main(void)

{

struct node \*start=NULL;

start=create\_list(start);

display(start);

MoveSmall(start);

display(start);

return 0;

}/\*End of main()\*/

void MoveSmall(struct node \*start)

{

struct node \*p,\*q;

int tmp;

if(start==NULL)

return;

p=start;

q=start->link;

while(q!=NULL)

{

if(p->info > q->info )

{

tmp = p->info;

p->info = q->info;

q->info = tmp;

}

q=q->link;

}

}/\*End of MoveSmall()\*/

struct node \*create\_list(struct node \*start)

{

int i,n,data;

printf("Enter the number of nodes : ");

scanf("%d",&n);

start=NULL;

if(n==0)

return start;

printf("Enter the element to be inserted : ");

scanf("%d",&data);

start=addatbeg(start,data);

for(i=2;i<=n;i++)

{

printf("Enter the element to be inserted : ");

scanf("%d",&data);

start=addatend(start,data);

}

return start;

}/\*End of create\_list()\*/

void display(struct node \*start)

{

struct node \*p;

if(start==NULL)

{

printf("List is empty\n");

return;

}

p=start;

printf("List is :\n");

while(p!=NULL)

{

printf("%d ",p->info);

p=p->link;

}

printf("\n\n");

}/\*End of display() \*/

struct node \*addatbeg(struct node \*start,int data)

{

struct node \*tmp;

tmp=(struct node \*)malloc(sizeof(struct node));

tmp->info=data;

tmp->link=start;

start=tmp;

return start;

}/\*End of addatbeg()\*/

struct node \*addatend(struct node \*start,int data)

{

struct node \*p,\*tmp;

tmp=(struct node \*)malloc(sizeof(struct node));

tmp->info=data;

p=start;

while(p->link!=NULL)

p=p->link;

p->link=tmp;

tmp->link=NULL;

return start;

}/\*End of addatend()\*/

21.

#include<stdio.h>

#include<stdlib.h>

struct node

{

int info;

struct node \*link;

};

struct node \*create\_list(struct node \*start);

void display(struct node \*start);

struct node \*addatbeg(struct node \*start,int data);

struct node \*addatend(struct node \*start,int data);

struct node \*RemoveFirstInsertLast(struct node \*start);

int main(void)

{

struct node \*start=NULL;

start=create\_list(start);

display(start);

start = RemoveFirstInsertLast(start);

display(start);

return 0;

}/\*End of main()\*/

struct node \*RemoveFirstInsertLast(struct node \*start)

{

struct node \*p;

p=start;

if(start==NULL || start->link==NULL)/\*list empty or only one element in the list\*/

return start;

while(p->link!=NULL)

p=p->link;

/\*Now p points to last pointer\*/

p->link = start;

start = start->link;

p->link->link=NULL;

return start;

}/\*End of RemoveFirstInsertLast()\*/

struct node \*create\_list(struct node \*start)

{

int i,n,data;

printf("Enter the number of nodes : ");

scanf("%d",&n);

start=NULL;

if(n==0)

return start;

printf("Enter the element to be inserted : ");

scanf("%d",&data);

start=addatbeg(start,data);

for(i=2;i<=n;i++)

{

printf("Enter the element to be inserted : ");

scanf("%d",&data);

start=addatend(start,data);

}

return start;

}/\*End of create\_list()\*/

22.

#include<stdio.h>

#include<stdlib.h>

struct node

{

int info;

struct node \*link;

};

struct node \*create\_list(struct node \*start);

void display(struct node \*start);

struct node \*addatbeg(struct node \*start,int data);

struct node \*addatend(struct node \*start,int data);

struct node \*RemoveLastInsertFirst(struct node \*start);

int main(void)

{

struct node \*start=NULL;

start=create\_list(start);

display(start);

start = RemoveLastInsertFirst(start);

display(start);

return 0;

}/\*End of main()\*/

struct node \*RemoveLastInsertFirst(struct node \*start)

{

struct node \*p;

p=start;

if(start==NULL || start->link==NULL )/\*list empty or only one element in the list\*/

return start;

while(p->link->link!=NULL)

p=p->link;

/\*Now p points to second last node\*/

p->link->link=start;

start=p->link;

p->link=NULL;

return start;

}/\*End of RemoveLastInsertFirst()\*/

struct node \*create\_list(struct node \*start)

{

int i,n,data;

printf("Enter the number of nodes : ");

scanf("%d",&n);

start=NULL;

if(n==0)

return start;

printf("Enter the element to be inserted : ");

scanf("%d",&data);

start=addatbeg(start,data);

for(i=2;i<=n;i++)

{

printf("Enter the element to be inserted : ");

scanf("%d",&data);

start=addatend(start,data);

}

return start;

}/\*End of create\_list()\*/

void display(struct node \*start)

{

struct node \*p;

if(start==NULL)

{

printf("List is empty\n");

return;

}

p=start;

printf("List is :\n");

while(p!=NULL)

{

printf("%d ",p->info);

p=p->link;

}

printf("\n\n");

}/\*End of display() \*/

struct node \*addatbeg(struct node \*start,int data)

{

struct node \*tmp;

tmp=(struct node \*)malloc(sizeof(struct node));

tmp->info=data;

tmp->link=start;

start=tmp;

return start;

}/\*End of addatbeg()\*/

struct node \*addatend(struct node \*start,int data)

{

struct node \*p,\*tmp;

tmp=(struct node \*)malloc(sizeof(struct node));

tmp->info=data;

p=start;

while(p->link!=NULL)

p=p->link;

p->link=tmp;

tmp->link=NULL;

return start;

}/\*End of addatend()\*/

23.

#include<stdio.h>

#include<stdlib.h>

struct node

{

int info;

struct node \*link;

};

struct node \*create\_list(struct node \*start);

void display(struct node \*ptr);

void Rdisplay(struct node \*ptr);

int length(struct node \*ptr);

int sum (struct node \*ptr);

int search(struct node \*ptr, int item );

struct node \*insertLast(struct node \*ptr, int value);

struct node \*delLast(struct node \*ptr );

struct node \*reverse(struct node \*ptr);

int main(void)

{

struct node \*start=NULL;

int choice,data;

while(1)

{

printf("1.Create List\n");

printf("2.Display\n");

printf("3.Display in reverse order\n");

printf("4.Count\n");

printf("5.Sum of elements\n");

printf("6.Search\n");

printf("7.Insert at last\n");

printf("8.Delete the last node\n");

printf("9.Reverse the list\n");

printf("10.Quit\n");

printf("Enter your choice : ");

scanf("%d",&choice);

printf("\n");

switch(choice)

{

case 1:

start=create\_list(start);

break;

case 2:

display(start);

printf("\n\n");

break;

case 3:

Rdisplay(start);

printf("\n\n");

break;

case 4:

printf("Number of elements = %d\n\n",length(start));

break;

case 5:

printf("Sum of elements = %d\n\n",sum(start));

break;

case 6:

printf("Enter the element to be searched : ");

scanf("%d",&data);

if( search(start,data) == 1 )

printf("Element present\n\n");

else

printf("Element not present\n\n");

break;

case 7:

printf("Enter the element to be inserted : ");

scanf("%d",&data);

start=insertLast(start,data);

break;

case 8:

start=delLast(start);

printf("Last node deleted......\n");

break;

case 9:

start=reverse(start);

break;

case 10:

exit(1);

default:

printf("Wrong choice\n");

}/\*End of switch \*/

}/\*End of while \*/

return 0;

}/\*End of main()\*/

struct node \*create\_list(struct node \*start)

{

int i,n,value;

struct node \*q,\*tmp;

printf("Enter the number of nodes : ");

scanf("%d",&n);

start=NULL;

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

{

printf("Enter the element to be inserted : ");

scanf("%d",&value);

tmp= malloc(sizeof(struct node));

tmp->info=value;

tmp->link=NULL;

if(start==NULL) /\*If list is empty \*/

start=tmp;

else

{ /\*Element inserted at the end \*/

q=start;

while(q->link!=NULL)

q=q->link;

q->link=tmp;

}

}

return start;

}/\*End of create\_list()\*/

void display(struct node \*ptr)

{

if(ptr==NULL)

return;

printf("%d ",ptr->info);

display(ptr->link);

}/\*End of display()\*/

void Rdisplay(struct node \*ptr)

{

if(ptr==NULL)

return;

Rdisplay(ptr->link);

printf("%d ",ptr->info);

}/\*End of Rdisplay()\*/

int length(struct node \*ptr)

{

if(ptr==NULL)

return 0;

return 1 + length(ptr->link);

}/\*End of length()\*/

int sum (struct node \*ptr)

{

if (ptr == NULL)

return 0;

return ptr->info + sum(ptr->link);

}/\*End of sum()\*/

int search(struct node \*ptr, int item)

{

if(ptr==NULL)

return 0;

if(ptr->info == item)

return 1;

return search(ptr->link, item);

}/\*End of search()\*/

struct node \*insertLast(struct node \*ptr, int item)

{

struct node \*temp;

if (ptr == NULL)

{

temp = malloc(sizeof(struct node));

temp->info = item;

temp->link = NULL;

return temp;

}

ptr->link = insertLast(ptr->link, item);

return ptr;

}/\*End of insertLast()\*/

struct node \*delLast(struct node \*ptr )

{

if( ptr->link == NULL )

{

free(ptr);

return NULL;

}

ptr->link = delLast(ptr->link);

return ptr;

}/\*End of delLast()\*/

struct node \*reverse(struct node \*ptr)

{

struct node \*temp;

if( ptr->link == NULL )

return ptr;

temp=reverse(ptr->link);

ptr->link->link=ptr;

ptr->link=NULL;

return temp;

}/\*End of reverse()\*/

Exercise-Files 12

1.

#include<stdio.h>

int main(void)

{

FILE \*fptr;

unsigned char ch;

fptr=fopen("myfile.txt","r");

while((ch=fgetc(fptr))!=EOF)

putchar(ch);

fclose(fptr);

return 0;

}

2.

#include<stdio.h>

int main(void)

{

FILE \*fp;

int ch;

fp=fopen("myfile.txt","w");

fprintf(fp,"If equal love there cannot be..");

fputc(26,fp);

fprintf(fp,"..let the more loving one be me\n");

fclose(fp);

fp=fopen("myfile.txt","r");

while((ch=fgetc(fp))!=EOF)

putchar(ch);

return 0;

}

3.

#include<stdio.h>

#include<stdlib.h>

int main(void)

{

FILE \*fptr1,\*fptr2;

char fname[20];

printf("Enter the path of first file : ");

scanf("%s",fname);

fptr1=fopen(fname,"r");

if(fptr1==NULL)

{

printf("Error in opening first file\n");

exit(1);

}

fptr2=fopen("c:\mydir\names.txt","r");

if(fptr2==NULL)

{

printf("Error in opening second file\n");

exit(1);

}

fclose(fptr1);

fclose(fptr2);

return 0;

}

4.

#include<stdio.h>

int main(void)

{

FILE \*fptr;

int ch;

fptr = fopen("names.txt",'r');

while((ch=fgetc(fptr))!=EOF)

putchar(ch);

fclose(fptr);

return 0;

}

5.

#include<stdio.h>

int main(void)

{

char name[50];

int empid;

fprintf(stdout,"Enter your name : ");

fgets(name,50,stdin);

fprintf(stdout,"Enter your empid : ");

fscanf(stdin,"%d",&empid);

fprintf(stdout,"Your empid is : %d",empid);

fputc('\n',stdout);

fprintf(stdout,"Your name is : ");

fputs(name,stdout);

return 0;

}

6.

#include<stdio.h>

int main(void)

{

FILE \*fptr;

char str[80];

fptr = fopen("test.txt","r");

while(fgets(str,80,fptr)!=EOF)

puts(str);

return 0;

}

7.

#include<stdio.h>

#include<stdlib.h>

int main(int argc,char \*argv[])

{

int i,sum=0;

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

sum=sum+atoi(argv[i]);

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

return 0;

}

7.

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

int main(void)

{

FILE \*fp1,\*fp2;

char str[100];

if((fp1=fopen("source.txt","r"))==NULL)

{

printf("Can’t open source file\n");

exit(1);

}

if((fp2=fopen("dest.txt","w"))==NULL)

{

printf("Can’t open destination file\n");

exit(1);

}

while((fgets(str,100,fp1)) != NULL)

if(strcmp(str,"\n")!=0)

fputs(str, fp2);

fclose(fp1);

fclose(fp2);

return 0;

}

8.

#include<stdio.h>

#include<stdlib.h>

#include<ctype.h>

int main(void)

{

FILE \*fp1,\*fp2;

int c;

char fname[20];

printf("Enter name of the file : ");

scanf("%s",fname);

if((fp1=fopen(fname,"r"))==NULL)

{

printf("Error in opening file\n");

exit(1);

}

if((fp2=fopen("tempfile.txt","w"))==NULL)

{

printf("Error in opening file\n");

exit(1);

}

while((c=fgetc(fp1))!=EOF)

fputc(toupper(c),fp2);

fclose(fp1);

fclose(fp2);

remove(fname);

rename("tempfile.txt",fname);

return 0;

}

9.

#include<stdio.h>

#include<stdlib.h>

#include<ctype.h>

int main(void)

{

FILE \*fp1,\*fp2;

int c;

char fname[20];

printf("Enter name of the file : ");

scanf("%s",fname);

if((fp1=fopen(fname,"r"))==NULL)

{

printf("Error in opening file\n");

exit(1);

}

if((fp2=fopen("tempfile.txt","w"))==NULL)

{

printf("Error in opening file\n");

exit(1);

}

while((c=fgetc(fp1))!=EOF)

fputc(toupper(c),fp2);

fclose(fp1);

fclose(fp2);

remove(fname);

rename("tempfile.txt",fname);

return 0;

}

10.

#include<stdio.h>

#include<stdlib.h>

#include<ctype.h>

int main(void)

{

FILE \*p;

char ch, fname[50];

int n=0;

printf("Enter name of the file : ");

scanf("%s",fname);

if((p=fopen(fname,"r"))==NULL)

{

printf("Error in opening file\n");

exit(1);

}

while((ch=fgetc(p))!=EOF)

{

if(isalnum(ch))

n++;

}

printf("Total alphanumeric characters = %d\n",n);

fclose(p);

return 0;

}

11.

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

int main(void)

{

FILE \*fp1,\*fp2;

char str[100],fname[20];

int n=1,page=1;

printf("Enter name of the file : ");

scanf("%s",fname);

if((fp1=fopen(fname,"r"))==NULL)

{

printf("Error in opening file\n");

exit(1);

}

if((fp2=fopen("tempfile.txt","w"))==NULL)

{

printf("Error in opening file\n");

exit(1);

}

while((fgets(str,100,fp1))!=NULL)

{

if(n%50==1)

fprintf(fp2,"\n......Page %d.....\n",page++);

fprintf(fp2,"%d ",n++);

fputs(str, fp2);

}

fclose(fp1);

fclose(fp2);

remove(fname);

rename("tempfile.txt",fname);

return 0;

}

12.

#include<stdio.h>

#include<stdlib.h>

int main(void)

{

FILE \*fp1,\*fp2;

char name[50];

int c1,c2,found='n';

printf("Enter the file name : ");

scanf("%s",name);

if((fp1=fopen(name,"r"))==NULL)

{

printf("Error in opening file\n");

exit(1);

}

fp2 = fopen("new.c","w");

c1 = fgetc(fp1);

c2 = fgetc(fp1);

while(c2!=EOF)

{

if(c1=='/' && c2=='/')

found = 'y';

if(found=='n')

fputc(c1,fp2);

if(c2=='\n')

{

fputc('\n',fp2);

found = 'n';

c2 = fgetc(fp1);

}

c1 = c2;

c2 = fgetc(fp1);

}

fclose(fp1);

fclose(fp2);

return 0;

}

13.

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

int main(int argc,char \*argv[])

{

int i,c;

FILE \*fptr1,\*fptr2;

if((fptr2=fopen(argv[1],"w"))==NULL)

{

printf("Error in opening file\n");

exit(1);

}

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

for(i=2; i<argc; i++)

{

if((fptr1=fopen(argv[i],"r"))==NULL)

{

printf("Error in opening file\n");

exit(1);

}

while((c=fgetc(fptr1))!=EOF)

fputc(c,fptr2);

fclose(fptr1);

}

fclose(fptr2);

return 0;

}

14.

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

struct employee

{

char name[20];

int age;

int sal;

};

int main(void)

{

struct employee e,temp,emp[50];

FILE \*fp;

int i,j,k=0,n;

fp = fopen("emp","wb+");

if(fp==NULL)

{

printf("Error in opening file\n");

exit(1);

}

/\*Enter records in the file\*/

printf("Enter number of records : ");

scanf("%d",&n);

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

{

printf("Enter name : ");

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

printf("Enter age : ");

scanf("%d",&e.age);

printf("Enter salary : ");

scanf("%d",&e.sal);

fwrite(&e,sizeof(e),1,fp);

}

/\*Read records from the file and store in the array emp\*/

rewind(fp);

while(fread(&e,sizeof(e),1,fp)==1)

emp[k++] = e;

/\*Sort the array emp\*/

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

{

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

{

if(strcmp(emp[j].name,emp[j+1].name)>0)

{

temp=emp[j];

emp[j]=emp[j+1];

emp[j+1]=temp;

}

else if(strcmp(emp[j].name, emp[j+1].name)==0)

if(emp[j].sal > emp[j+1].sal)

{

temp=emp[j];

emp[j]=emp[j+1];

emp[j+1]=temp;

}

}

}

/\*Write the sorted array emp in the file\*/

rewind(fp);

fwrite(emp,sizeof(e),k,fp);

/\*Read the file and display the records\*/

rewind(fp);

while(fread(&e,sizeof(e),1,fp)==1)

{

printf("%s\t",e.name);

printf("%d\t",e.age);

printf("%d\n",e.sal);

}

fclose(fp);

return 0;

}

Exercise-The C Preeprocessor

1.

#define MAX 5;

#include<stdio.h>

int main(void)

{

printf(" %d ",MAX);

return 0;

}

2.

#include<stdio.h>

#define MSSG printf("If you lapse,don't collapse\n");

int main(void)

{

MSSG

return 0;

}

3.

#include<stdio.h>

#define PROD (x,y) ((x)\*(y))

int main(void)

{

int a=3,b=4;

printf("Product of a and b = %d",PROD(a,b));

return 0;

}

4.

#include<stdio.h>

#define A 50

#define B A+100

int main(void)

{

int i,j;

i=B/20;

j=500-B;

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

return 0;

}

5.

#include<stdio.h>

#define NEW\_LINE printf("\n");

#define BLANK\_LINES(n) { int i; for(i=0; i<n; i++) printf("\n"); }

int main(void)

{

printf("When you have a chance");

NEW\_LINE

printf("to embrace an opportunity");

BLANK\_LINES(3)

printf("Give it a big hug");

NEW\_LINE

return 0;

}

6.

#include<stdio.h>

#define INFINITE while(1)

#define CHECK(a) if(a==0) break

int main(void)

{

int x=5;

INFINITE

{

printf("%d ",x--);

CHECK(x);

}

return 0;

}

7.

#include<stdio.h>

#define ABS(x) ((x)<0? -(x) : (x))

int main(void)

{

int array[4]={1,-2,3,-4};

int \*p=array+3;

while(p>=array)

{

printf("%d ",ABS(\*p));

p--;

}

return 0;

}

8.

#include<stdio.h>

#define . ;

int main(void)

{

printf("If the lift to success is broken, ").

printf("Try the stairs.").

return 0;

}

9.

#include<stdio.h>

#define CUBE(x) (x\*x\*x)

int main(void)

{

printf("%d\n",CUBE(1+2));

return 0;

}

10.

#include<stdio.h>

#define CUBE(x) ((x)\*(x)\*(x))

int main(void)

{

int i=1;

while(i<=8)

printf("%d\t",CUBE(i++));

return 0;

}

11.

#include<stdio.h>

#define SWAP(dtype,x,y) { dtype t; t=x+y, x=t-x, y=t-y; }

int main(void)

{

int a=1, b=2, x=3, y=4, s=25, t=26;

SWAP(int,a,b)

SWAP(int,x,y)

SWAP(int,s,t)

printf("a=%d, b=%d, x=%d, y=%d, s=%d, t=%d\n",a,b,x,y,s,t);

return 0;

}

12.

#include<stdio.h>

#define INC(dtype,x,i) x=x+i

int main(void)

{

int arr[5]={20,34,56,12,96},\*ptr=arr;

INC(int,arr[2],3);

INC(int\*,ptr,2);

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

return 0;

}

13.

#include<stdio.h>

#define INT int

int main(void)

{

INT a=2,\*p=&a;

printf("%d %d\n",a,\*p);

return 0;

}

14.

#include<stdio.h>

#define Y 10

int main(void)

{

#if X || Y && Z

printf("Sea in Depth\n");

#else

printf("See in depth\n");

#endif

return 0;

}

15.

#include<stdio.h>

int main(void)

{

int x=3,y=4,z;

z=x+y;

#include<string.h>

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

return 0;

}

16.

#include<stdio.h>

#define DIFF(FNAME, DTYPE, RTYPE) \

RTYPE FNAME(DTYPE X, DTYPE Y){ return X-Y;}

DIFF(diff\_int, int, int)

DIFF(diff\_iptr, int\*, int)

DIFF(diff\_float, float, float);

DIFF(diff\_fptr, float\*, int);

int main(void)

{

int iarr[5] = {1,2,3,4,5},a,p,q;

float farr[7] = {1.2,2.3,3.4,4.5,5.6,6.7,7.8},b;

a = diff\_int(iarr[4],iarr[1]);

b = diff\_float(farr[6],farr[2]);

p = diff\_iptr(&iarr[4],&iarr[1]);

q = diff\_fptr(&farr[4],&farr[1]);

printf("a=%d, b=%.1f, p=%d, q=%d\n",a,b,p,q);

return 0;

}

17.

#include<stdio.h>

#define MAX 3

int main(void)

{

printf("Value of MAX is %d\n",MAX);

#undef MAX

#ifdef MAX

printf("Have a good day");

#endif

return 0;

}

18.

#include<stdio.h>

#define PRINT1(message) printf(message);

#define PRINT2(message) printf("message");

#define PRINT3(message) printf(#message);

int main(void)

{

PRINT1("If we rest, we rust.\n")

PRINT2("Attack life, its going to kill you anyways.\n")

PRINT3("Well done is better than well said.\n")

return 0;

}

19.

#include<stdio.h>

#define show(value) printf(#value" = %d\n",value);

int main(void)

{

int a=10,b=5,c=4;

show(a/b\*c);

return 0;

}

20.

#include<stdio.h>

#define MACRO(a) if(a<=5) printf(#a" = %d\n",a);

int main(void)

{

int x=6, y=15;

if(x<=y)

MACRO(x);

else

MACRO(y);

return 0;

}

21.

#include<stdio.h>

int main(void)

{

#line 100 "system.c"

printf("%d %s\n",\_\_LINE\_\_,\_\_FILE\_\_);

return 0;

}

22.

#include<stdio.h>

#define toupper(c)\

((c)>='a' && (c)<='z' ? (c) +('A'-'a'):(c))

int main(void)

{

char str[] = "Devanshi", \*p;

p=str;

while(\*p!='\0')

printf("%c",\*p++);

printf("\n");

p=str;

while(\*p!='\0')

printf("%c",toupper(\*p++));

return 0;

}

Exercise- Operations on Bits 14

1.

#include<stdio.h>

int main(void)

{

int x=7,y=19;

printf("%d %d ",x&y,x&&y);

printf("%d %d ",x|y,x||y);

printf("%d\n",x^y);

return 0;

}

2.

#include<stdio.h>

void displayBits(int x);

int main(void)

{

unsigned int x,y,z;

displayBits(0xFFFF);

x=y=z=0xFFFF;

x=(x>>5)<<5; displayBits(x);

y=(y>>3)<<3; displayBits(y);

z=(z>>2)<<2; displayBits(z);

return 0;

}

void displayBits(int x)

{

int i,mask;

for(i=31; i>=0; i--)

{

mask = 1<<i;

putchar((x & mask)?'1':'0'); /\*Test and print ith bit\*/

if(i%8==0)

putchar(' '); /\*Space after 8 bits\*/

}

printf("\n");

}

3.

#include<stdio.h>

int main(void)

{

int k;

k=((3<<4)^(96>>1));

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

return 0;

}

4.

#include<stdio.h>

int main(void)

{

int x=0x1F;

x<<2;

printf("%X ",x);

x>>2;

printf("%X\n",x);

return 0;

}

5.

#include<stdio.h>

int main(void)

{

unsigned int arr\_mask[]={0x1, 0x2, 0x4, 0x8, 0x10, 0x20, 0x40, 0x80, 0x100, 0x200,

0x400, 0x800, 0x1000,0x2000, 0x4000, 0x8000, 0x10000, 0x20000, 0x40000, 0x80000,

0x100000, 0x200000,0x400000, 0x800000, 0x1000000, 0x2000000, 0x4000000, 0x8000000,

0x10000000, 0x20000000, 0x40000000, 0x80000000};

int i,x=0x54038F;

for(i=31; i>=0; i--)

x&arr\_mask[i] ? putchar('1'): putchar('0');

return 0;

}

6.

#include<stdio.h>

int main(void)

{

unsigned int num=0xA01D,pos=3,bit;

unsigned int mask=1<<pos;

bit=(num&mask)>>pos;

printf("%u\n", bit);

return 0;

}

7.

#include<stdio.h>

int main(void)

{

unsigned int num=0x1F,pos=3,bit;

bit=(num>>pos)&1;

printf("%u\n",bit);

return 0;

}

8.

#include<stdio.h>

int main(void)

{

int i,num=0xA0DF;

for(i=31; i>=0; i--)

printf("%d",(num>>i)&1);

return 0;

}

9.

#include<stdio.h>

int main(void)

{

int i,num=0x1A3B;

unsigned int mask=1<<31;

for(i=31; i>=0; i--)

{

(num & mask) ? printf("1") : printf("0");

mask=mask>>1;

}

return 0;

}

10.

#include<stdio.h>

void func(int x);

int main(void)

{

func(0x1AE3);

}

void func(int x)

{

int i,mask;

mask=1<<31;

for(i=1; i<=32; i++)

{

putchar((x&mask)?'1':'0');

x<<=1;

if(i%8==0)

putchar(' ');

}

printf("\n");

return 0;

}

11.

#include<stdio.h>

int func(unsigned int x);

int main(void)

{

printf("%d\n",func(0x1AE3));

return 0;

}

int func(unsigned x)

{

int count=0,mask=1,i;

for(i=0; i<32; i++)

{

if((x&mask)!=0) /\*Check ith bit\*/

count++;

mask<<=1;

}

return count;

}

12.

#include<stdio.h>

unsigned int func(unsigned int x);

void displayBits(int x);

int main(void)

{

unsigned int x=0x1AE3;

displayBits(x);

x=func(x);

printf("%X\n",x);

displayBits(x);

return 0;

}

unsigned int func(unsigned int num)

{

unsigned int i,r=0;

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

{

r=(r<<1)|num & 1;

num>>=1;

}

r<<=32-i;

return r;

}

void displayBits(int x)

{

int i,mask;

for(i=31; i>=0; i--)

{

mask = 1<<i;

putchar((x & mask)?'1':'0'); /\* Test and print ith bit\*/

if(i%8==0)

putchar(' '); /\*Space after 8 bits\*/

}

printf("\n");

}

13.

#include<stdio.h>

unsigned int func(unsigned int x);

void displayBits(int x);

int main(void)

{

unsigned int x=0x1AE3;

displayBits(x);

x=func(x);

printf("%X\n",x);

displayBits(x);

return 0;

}

unsigned int func(unsigned int num)

{

unsigned int lmask,rmask,mask;

lmask=1<<31;

rmask=1;

while(lmask > rmask)

{

mask=lmask|rmask;

if((num&mask)!=0 && (num&mask)!=mask)

num^=mask;

lmask>>=1;

rmask<<=1;

}

return num;

}

void displayBits(int x)

{

int i,mask;

for(i=31; i>=0; i--)

{

mask = 1<<i;

putchar((x & mask)?'1':'0'); /\* Test and print ith bit\*/

if(i%8==0)

putchar(' '); /\*Space after 8 bits\*/

}

printf("\n");

}

14.

#include<stdio.h>

void displayBits(int x);

int main(void)

{

unsigned int x=0x123F4;

displayBits(x);

printf("(i) Set most significant bit\n");

x|=(1<<31); displayBits(x);

printf("(ii) Clear most significant bit\n");

x&=~(1<<31); displayBits(x);

printf("(iii) Invert all bits\n");

x^=~0; displayBits(x);

printf("(iv) Set all bits\n");

x|=~0; displayBits(x);

printf("(v) Invert least significant byte\n");

x^=0xFF; displayBits(x);

printf("(vi) Invert all bits at even positions 0,2,4,6,8..\n");

x^=0x55555555;displayBits(x);

printf("(vii) Invert all bits at odd positions 1,3,5,7,9..\n");

x^=0xAAAAAAAA;displayBits(x);

printf("(viii) Clear all bits at even positions 0,2,4,6,8..\n");

x&=0xAAAAAAAA;displayBits(x);

printf("(ix) Clear all bits at odd positions 1,3,5,7,9..\n");

x&=0x55555555;displayBits(x);

printf("\n\n");

x=0x123FF;

displayBits(x);

printf("(x)Insert 3 trailing zeros \n");

x=(x>>3)<<3; displayBits(x);

printf("(xi) Find if every bit is set \n");

(x^~0) ? printf("Not all Set\n"): printf("All Set\n");

(x&~0)==~0 ? printf("All Set\n"): printf("Not all Set\n");

printf("\n\n");

x=4;

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

printf("(xii) Multiply by 7\n");

x=(x<<3)-x;

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

printf("(xiii) Multiply by 9\n");

x=(x<<3)+x;

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

printf("(xiv) Multiply by 3.5\n");

x=((x<<3)-x)>>1;

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

return 0;

}

void displayBits(int x)

{

int i,mask;

for(i=31; i>=0; i--)

{

mask = 1<<i;

putchar((x & mask)?'1':'0'); /\* Test and print ith bit\*/

if(i%8==0)

putchar(' '); /\*Space after 8 bits\*/

}

printf("\n");

}

15.

#include<stdio.h>

int wordlength(void);

int main(void)

{

printf("%d ",wordlength());

return 0;

}

int wordlength(void)

{

unsigned x;

int count=0;

for(x=~0; x!=0; x=x<<1)

count++;

return count;

}

16.

#include<stdio.h>

int main(void)

{

unsigned x=1434,y=32,r;

r=x&(y-1);

printf("%d %d",x%y,r);

return 0;

}

17.

#include<stdio.h>

int isMultiple(int x,int n);

int main(void)

{

int x,i;

printf("Enter x and i : ");

scanf("%d%d",&x,&i);

if(isMultiple(x,i) )

printf("%d is multiple of 2 to the power of %d\n",x,i);

else

printf("%d is not multiple of 2 to the power of %d\n",x,i);

return 0;

}

int isMultiple(int x,int i)

{

return !(x & (~(~0<<i)));

}

18.

#include<stdio.h>

void displayBits(int x);

int count\_setbits2(unsigned x);

int main(void)

{

unsigned x=1034,y=1083;

displayBits(x);

displayBits(y);

printf("%d\n",count\_setbits2(x^y));

return 0;

}

int count\_setbits2(unsigned x)

{

int count=0;

while(x!=0)

{

count++;

x=x&(x-1);

}

return count;

}

void displayBits(int x)

{

int i,mask;

for(i=31; i>=0; i--)

{

mask=1<<i;

putchar((x & mask)?'1':'0'); /\* Test and print ith bit\*/

if(i%8==0)

putchar(' '); /\*Space after 8 bits\*/

}

printf("\n");

}

19.

#include<stdio.h>

void displayBits(int x);

int main(void)

{

int count,x=0x1F,y=0xF1,z;

displayBits(x);

displayBits(y);

count=0;

for(z=x^y; z!=0; z=z&(z-1))

count++;

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

count=0;

for(z=x^y; z!=0; z>>=1)

count+=z&1;

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

return 0;

}

void displayBits(int x)

{

int i,mask;

for(i=31; i>=0; i--)

{

mask = 1<<i;

putchar((x & mask)?'1':'0'); /\* Test and print ith bit\*/

if(i%8==0)

putchar(' '); /\*Space after 8 bits\*/

}

printf("\n");

}

20.

#include<stdio.h>

void displayBits(int x);

#include<math.h>

int main(void)

{

int mask,p=7,n=4,x=103145;

displayBits(x);

if(n<0)

mask = ~(~(~0<<abs(n))<<p);

else

mask = ~(~(~0<<n)<<p-n+1);

displayBits(mask);

x=x&mask;

displayBits(x);

p=7,n=-4,x=103145;

displayBits(x);

if(n<0)

mask = ~(~(~0<<abs(n))<<p);

else

mask = ~(~(~0<<n)<<p-n+1);

displayBits(mask);

x=x&mask;

displayBits(x);

return 0;

}

void displayBits(int x)

{

int i,mask;

for(i=31; i>=0; i--)

{

mask = 1<<i;

putchar((x & mask)?'1':'0'); /\* Test and print ith bit\*/

if(i%8==0)

putchar(' '); /\*Space after 8 bits\*/

}

printf("\n");

}

21.

#include<stdio.h>

void displayBits(int x);

int main(void)

{

unsigned int x=103145, p=7, n=5;

displayBits(x);

x=func(x,p,n);

displayBits(x);

return 0;

}

unsigned int func(unsigned int x, int p, int n)

{

return (x>>(p+1-n)) & ~(~0<<n) ;

}

void displayBits(int x)

{

int i,mask;

for(i=31; i>=0; i--)

{

mask = 1<<i;

putchar((x & mask)?'1':'0'); /\* Test and print ith bit\*/

if(i%8==0)

putchar(' '); /\*Space after 8 bits\*/

}

printf("\n");

}

22.

#include<stdio.h>

void displayBits(int x);

int main(void)

{

unsigned x,y,r,mask,i=9;

x=0x123;

y=0xffffff;

mask=~0<<i+1;

r=(x&~mask) | (y&mask);

displayBits(x);

displayBits(y);

displayBits(r);

return 0;

}

void displayBits(int x)

{

int i,mask;

for(i=31; i>=0; i--)

{

mask = 1<<i;

putchar((x & mask)?'1':'0'); /\* Test and print ith bit\*/

if(i%8==0)

putchar(' '); /\*Space after 8 bits\*/

}

printf("\n");

}

23.

#include<stdio.h>

void displayBits(int x);

unsigned clear(unsigned x,int i,int j);

int main(void)

{

unsigned x = 0x23173b4;

displayBits(x);

x = clear(x,3,7) ;

displayBits(x);

return 0;

}

unsigned clear(unsigned x,int i,int j)

{

unsigned mask = ~0<<(j+1) | (1<<i)-1;

return x&mask;

}

void displayBits(int x)

{

int i,mask;

for(i=31; i>=0; i--)

{

mask = 1<<i;

putchar((x & mask)?'1':'0'); /\* Test and print ith bit\*/

if(i%8==0)

putchar(' '); /\*Space after 8 bits\*/

}

printf("\n");

}

24.

#include<stdio.h>

unsigned set(int x,int i,int j);

void displayBits(int x);

int main(void)

{

unsigned x=0x23172b0;

displayBits(x);

x=set(x,3,7);

displayBits(x);

return 0;

}

unsigned set(int x,int i,int j)

{

int mask=0,p;

for(p=i; p<j; p++)

mask=mask | 1<<p;

return x|mask;

}

void displayBits(int x)

{

int i,mask;

for(i=31; i>=0; i--)

{

mask = 1<<i;

putchar((x & mask)?'1':'0'); /\* Test and print ith bit\*/

if(i%8==0)

putchar(' '); /\*Space after 8 bits\*/

}

printf("\n");

}

25.

#include<stdio.h>

int main(void)

{

int x=511;

if((x&(x+1))==0)

printf("Yes\n");

else

printf("No\n");

return 0;

}

26.

#include<stdio.h>

int main(void)

{

int x=511;

if((x&(x+1))==0)

printf("Yes\n");

else

printf("No\n");

return 0;

}

27.

#include<stdio.h>

int main(void)

{

int x=245;

x=-(~x);

printf("%d\t", x);

x=~(-x);

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

return 0;

}

28.

#include<stdio.h>

void displayBits(int x);

int main(void)

{

int x=0x12E8;

displayBits(x);

x = x & ~(x-1);

displayBits(x);

x=0x12E8;

displayBits(x);

x = x & -x; /\*in two's complement machine\*/

displayBits(x);

return 0;

}

void displayBits(int x)

{

int i,mask;

for(i=31; i>=0; i--)

{

mask = 1<<i;

putchar((x & mask)?'1':'0'); /\* Test and print ith bit\*/

if(i%8==0)

putchar(' '); /\*Space after 8 bits\*/

}

printf("\n");

}

29.

#include<stdio.h>

#include<math.h>

unsigned func(unsigned int n);

unsigned func1(unsigned int n);

int main(void)

{

unsigned x=8;

printf("%d %d\n",func(x), func1(x));

x=35;

printf("%d %d\n",func(x), func1(x));

return 0;

}

unsigned func(unsigned int n)

{

int count=0;

while(n!=0)

{

n>>=1;

count++;

}

return count-1;

}

unsigned func1(unsigned int n)

{

return log(n)/log(2);

}

30.

#include<stdio.h>

void displayBits(int x);

unsigned parity(int x);

int main(void)

{

unsigned int n=67;

displayBits(n);

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

return 0;

}

unsigned parity(int x)

{

unsigned parity=0;

while(x)

{

parity=!parity;

x=x&(x-1);

}

return parity;

}

void displayBits(int x)

{

int i,mask;

for(i=31; i>=0; i--)

{

mask = 1<<i;

putchar((x & mask)?'1':'0'); /\* Test and print ith bit\*/

if(i%8==0)

putchar(' '); /\*Space after 8 bits\*/

}

printf("\n");

}

31.

#include<stdio.h>

unsigned mult(unsigned int x,unsigned int y);

int main(void)

{

printf("%d\n",mult(9,8));

return 0;

}

unsigned mult(unsigned int x,unsigned int y)

{

unsigned z=0;

while(y!=0)

{

if((y&1)!=0)

z=z+x;

x<<=1;

y>>=1;

}

return z;

}

32.

#include<stdio.h>

int main(void)

{

unsigned num=0xf000000f;

printf("%d\n",count1(num));

printf("%d\n",count2(num));

printf("%d\n",count3(num));

printf("%d\n",count4(num));

return 0;

}

int count1(unsigned x)

{

int count=0;

while(x!=0)

{

count++;

x=x&(x-1);

}

return count;

}

int count2(int x)

{

return 32-count1(x);

}

int count3(int x)

{

return count1(~x);

}

int count4(int x)

{

int c=0;

for(x=~x; x!=0; x=x&(x-1))

c++;

return c;

}

33.

#include<stdio.h>

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

void swap1(int \*a,int \*b);

int main(void)

{

int x=2,y=2;

int arr[5]={1,2,5,3,4},i,j,n=5,min;

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

{

min=i;

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

{

if(arr[min]>arr[j])

min=j ;

}

swap(&arr[i],&arr[min]);

/\*swap1(&arr[i],&arr[min]);\*/

}

printf("Sorted list is : \n");

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

printf("%d ",arr[i]);

printf("\n");

return 0;

}

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

{

int temp;

temp=\*a;

\*a=\*b;

\*b=temp;

}

void swap1(int \*x,int \*y)

{

\*x = \*x ^ \*y;

\*y = \*x ^ \*y;

\*x = \*x ^ \*y;

}

34.

#include<stdio.h>

void displayBits(int x);

int main(void)

{

unsigned n=123456;

displayBits(n);

n=n|(n-1);

displayBits(n);

return 0;

}

void displayBits(int x)

{

int i,mask;

for(i=31; i>=0; i--)

{

mask = 1<<i;

putchar((x & mask)?'1':'0'); /\* Test and print ith bit\*/

if(i%8==0)

putchar(' '); /\*Space after 8 bits\*/

}

printf("\n");

}

35.

#include<stdio.h>

void displayBits(int x);

int main(void)

{

unsigned n=0x1000010;

displayBits(n);

n|=n>>1;

n|=n>>2;

n|=n>>4;

n|=n>>8;

n|=n>>16;

displayBits(n);

return 0;

}

void displayBits(int x)

{

int i,mask;

for(i=31; i>=0; i--)

{

mask = 1<<i;

putchar((x & mask)?'1':'0'); /\* Test and print ith bit\*/

if(i%8==0)

putchar(' '); /\*Space after 8 bits\*/

}

printf("\n");

}

36.

#include<stdio.h>

void displayBits(int x);

unsigned nextHighestPow2(unsigned int n);

int main(void)

{

unsigned n=250,x;

displayBits(n);

x=nextHighestPow2(n);

displayBits(x);

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

return 0;

}

unsigned nextHighestPow2(unsigned int n)

{

n--;

n|=n>>1;

n|=n>>2;

n|=n>>4;

n|=n>>8;

n|=n>>16;

n++;

return n;

}

void displayBits(int x)

{

int i,mask;

for(i=31; i>=0; i--)

{

mask = 1<<i;

putchar((x & mask)?'1':'0'); /\* Test and print ith bit\*/

if(i%8==0)

putchar(' '); /\*Space after 8 bits\*/

}

printf("\n");

}

37.

#include<stdio.h>

int leadingZeros(int x);

int count\_setbits(x);

int rightProp(int n);

int main(void)

{

int x=0xFF;

printf("%d\n",leadingZeros(x));

return 0;

}

int leadingZeros(int x)

{

x=rightProp(x);

return count\_setbits(~x);

}

int rightProp(int n)

{

n|=n>>1;

n|=n>>2;

n|=n>>4;

n|=n>>8;

n|=n>>16;

return n;

}

int count\_setbits(int x)

{

int count=0;

while(x!=0)

{

count++;

x=x&(x-1);

}

return count;

}

38.

#include<stdio.h>

void displayBits(int x);

int func(int x);

int main(void)

{

int x=0x123;

displayBits(x);

x=func(x);

displayBits(x);

return 0;

}

int func(int x)

{

return x|(x+1);

}

void displayBits(int x)

{

int i,mask;

for(i=31; i>=0; i--)

{

mask = 1<<i;

putchar((x & mask)?'1':'0'); /\*Test and print ith bit\*/

if(i%8==0)

putchar(' '); /\*Space after 8 bits\*/

}

printf("\n");

}

39.

#include<stdio.h>

void displayBits(int x);

int main(void)

{

unsigned x=0x12E;

displayBits(x);

displayBits(swap(x));

return 0;

}

unsigned int swap(unsigned int x)

{

return ((x & 0x55555555)<<1) | ((x & 0xAAAAAAAA)>>1);

}

void displayBits(int x)

{

int i,mask;

for(i=31; i>=0; i--)

{

mask = 1<<i;

putchar((x & mask)?'1':'0'); /\* Test and print ith bit\*/

if(i%8==0)

putchar(' '); /\*Space after 8 bits\*/

}

printf("\n");

}

40.

#include<stdio.h>

void displayBits(int x);

unsigned swap4Bits(int x);

int main(void)

{

unsigned x=0x15F93A7;

displayBits(x);

displayBits(swap4Bits(x));

return 0;

}

unsigned swap4Bits(int x)

{

return ((x & 0x0F0F0F0F)<<4) | ((x & 0xF0F0F0F0)>>4);

}

void displayBits(int x)

{

int i,mask;

for(i=31; i>=0; i--)

{

mask = 1<<i;

putchar((x & mask)?'1':'0'); /\* Test and print ith bit\*/

if(i%8==0)

putchar(' '); /\*Space after 8 bits\*/

}

printf("\n");

}

41.

#include<stdio.h>

void displayBits(int x);

unsigned swapBytes(unsigned x);

int main(void)

{

unsigned x = 0x15F93A7;

displayBits(x);

displayBits(swapBytes(x));

return 0;

}

unsigned swapBytes(unsigned x)

{

return ((x & 0x00ff00ff) << 8) | ((x & 0xff00ff00) >> 8);

}

void displayBits(int x)

{

int i,mask;

for(i=31; i>=0; i--)

{

mask = 1<<i;

putchar((x & mask)?'1':'0'); /\* Test and print ith bit\*/

if(i%8==0)

putchar(' '); /\*Space after 8 bits\*/

}

printf("\n");

}

42.

#include<stdio.h>

void displayBits(int a);

unsigned reverseBytes(unsigned x);

int main(void)

{

unsigned x = 0xFA12CD04;

displayBits(x);

displayBits(reverseBytes(x));

return 0;

}

unsigned reverseBytes(unsigned x)

{

return (x>>24) | ((x & 0xFF0000)>>8) | ((x & 0xFF00)<<8) | (x<<24);

}

void displayBits(int x)

{

int i,mask;

for(i=31; i>=0; i--)

{

mask = 1<<i;

putchar((x & mask)?'1':'0'); /\* Test and print ith bit\*/

if(i%8==0)

putchar(' '); /\*Space after 8 bits\*/

}

printf("\n");

}

43.

#include<stdio.h>

void displayBits(int a);

unsigned func(int x);

int main(void)

{

unsigned x=0xFA2E4;

displayBits(x);

displayBits(func(x));

return 0;

}

unsigned func(int x)

{

x = ((x&0x55555555) << 1) | ((x&0xaaaaaaaa) >> 1);

x = ((x&0x33333333) << 2) | ((x&0xcccccccc) >> 2);

x = ((x&0x0f0f0f0f) << 4) | ((x&0xf0f0f0f0) >> 4);

x = ((x&0x00ff00ff) << 8) | ((x&0xff00ff00) >> 8);

x = ((x&0x0000ffff) << 16) | ((x&0xffff0000) >> 16);

return x;

}

void displayBits(int x)

{

int i,mask;

for(i=31; i>=0; i--)

{

mask = 1<<i;

putchar((x & mask)?'1':'0'); /\*Test and print ith bit\*/

if(i%8==0)

putchar(' '); /\*Space after 8 bits\*/

}

printf("\n");

}

44.

#include<stdio.h>

void displayBits(int a);

unsigned func(unsigned x);

int main(void)

{

unsigned x=12345678;

displayBits(x);

displayBits(func(x));

printf("%d\n",func(x));

return 0;

}

unsigned func(unsigned x)

{

x = (x & 0x55555555) + ((x & 0xaaaaaaaa) >> 1);

x = (x & 0x33333333) + ((x & 0xcccccccc) >> 2);

x = (x & 0x0f0f0f0f) + ((x & 0xf0f0f0f0) >> 4);

x = (x & 0x00ff00ff) + ((x & 0xff00ff00) >> 8);

x = (x & 0x0000ffff) + ((x & 0xffff0000) >> 16);

return x;

}

void displayBits(int x)

{

int i,mask;

for(i=31; i>=0; i--)

{

mask = 1<<i;

putchar((x & mask)?'1':'0'); /\*Test and print ith bit\*/

if(i%8==0)

putchar(' '); /\*Space after 8 bits\*/

}

printf("\n");

}

45.

#include<stdio.h>

void displayBits(int x);

int pack(int empid, int jobid, char jstatus, char gender, int age, char mstatus);

int main(void)

{

int emp;

int empid,jobid,age,mstatus;

char gender,jstatus;

emp=pack(2048,80,'P','F',50,3);

displayBits(emp);

/\*Unpack\*/

empid = emp & 0xFFF;

jobid = (emp & 0x7F000)>>12;

jstatus = (emp & 0x80000)>>19;

gender = (emp & 0x100000)>>20;

age = (emp & 0xFE00000)>>21;

mstatus = (emp & 0x30000000)>>28;

printf("%d\n%d\n%d\n%d\n%d\n%d\n",empid,jobid,jstatus,gender,age,mstatus);

return 0;

}

int pack(int empid, int jobid, char jstatus, char gender, int age, char mstatus)

{

int emp=0;

emp = emp | empid;

emp = emp | jobid<<12;

emp = emp | (jstatus == 'T' ? 0 : 1)<<19;

emp = emp | (gender == 'F' ? 0 : 1)<<20;

emp = emp | age<<21;

emp = emp | mstatus<<28;

return emp;

}

void displayBits(int x)

{

int i,mask;

for(i=31; i>=0; i--)

{

mask = 1<<i;

putchar((x & mask)?'1':'0'); /\* Test and print ith bit\*/

if(i%8==0)

putchar(' '); /\*Space after 8 bits\*/

}

printf("\n");

}

46.

#include<stdio.h>

void displayBits(int x);

int convertToBCD(int n);

int convertToBinary(int bcd);

int main(void)

{

int bcd,bin;

printf("Enter a number :");

scanf("%d",&bin);

displayBits(bin);

bcd=convertToBCD(bin);

displayBits(bcd);

bin=convertToBinary(bcd);

displayBits(bin);

return 0;

}

int convertToBCD(int n)

{

int rem,i,bcd=0;

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

{

rem=n%10; /\*taking last digit of number\*/

bcd = bcd | ( (rem & 0xF)<< i\*4 );

n/=10; /\*skipping last digit\*/

}

return bcd;

}

int convertToBinary(int bcd)

{

int i,bin=0,d=1;

for(i=0; i<32; i+=4)

{

bin+=d\*(bcd>>i & 0xF);

d\*=10;

}

return bin;

}

void displayBits(int x)

{

int i,mask;

for(i=31; i>=0; i--)

{

mask = 1<<i;

putchar((x & mask)?'1':'0'); /\* Test and print ith bit\*/

if(i%8==0)

putchar(' '); /\*Space after 8 bits\*/

}

printf("\n");

}

47.

#include<stdio.h>

char unpack(int n, int p);

unsigned pack\_chars1(char c1,char c2,char c3,char c4);

unsigned pack\_chars2(char c1,char c2,char c3,char c4);

int main(void)

{

int p1,p2;

p1 = pack\_chars1('p','q','r','s');

p2 = pack\_chars2('p','q','r','s');

printf("%c %c %c %c\n",unpack(p1,0),unpack(p1,1), unpack(p1,2), unpack(p1,3));

printf("%c %c %c %c\n",unpack(p2,0),unpack(p2,1), unpack(p2,2), unpack(p2,3));

return 0;

}

unsigned pack\_chars1(char c1,char c2,char c3,char c4)

{

int n;

n = c1;

n = n | c2<<8;

n = n | c3<<16;

n = n | c4<<24;

return n;

}

unsigned pack\_chars2(char c1,char c2,char c3,char c4)

{

int n;

n = c1;

n = (n<<8) | c2;

n = (n<<8) | c3;

n = (n<<8) | c4;

return n;

}

char unpack(int n, int p)

{

unsigned mask = 0xFF << p\*8;

return (n & mask)>>p\*8;

}

Exercise- Miscellaneous Features 15

1.

#include<stdio.h>

enum month{jan,feb,mar,apr,may};

int main(void)

{

enum month m;

m = ++feb;

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

return 0;

}

2.

#include<stdio.h>

enum day{sun=1,mon,tue,wed};

int main(void)

{

enum day d1;

printf("%d\t",mon);

d1 = mon+2;

printf("%d\n",d1);

return 0;

}

3.

#include<stdio.h>

struct tag

{

auto int x;

static int y;

};

int main(void)

{

struct tag s;

s.x=4;

s.y=5;

return 0;

}

4.

#include<stdio.h>

int var=6;

int main(void)

{

int var=18;

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

return 0;

}

5.

#include<stdio.h>

int main(void)

{

int i,sum=0;

for(i=0; i<5; i++)

{

int k=10;

sum = sum + k++;

}

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

return 0;

}

6.

#include<stdio.h>

int main(void)

{

int i,sum=0;

for(i=0; i<5; i++)

{

static int k=10;

sum = sum + k++;

}

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

return 0;

}

7.

#include<stdio.h>

int x=89;

void func1(int x);

void func2(void);

int main(void)

{

func1(x);

printf("%d\t",x);

func2();

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

return 0;

}

void func1(int x)

{

x++;

}

void func2(void)

{

x++;

}

8.

#include<stdio.h>

void func(void);

int x=2;

static int y=5;

int main(void)

{

int x=3;

func();

func();

printf("Inside main() : x=%d, y=%d\n",x,y);

return 0;

}

void func(void)

{

static int x;

x=x+2;

printf("Inside func(): x=%d, y=%d\n",x,y);

}

9.

#include<stdio.h>

void func1(void);

void func2(void);

int main(void)

{

func1();

func2();

return 0;

}

void func1(void)

{

extern int x;

x++;

printf("%d\t",x);

}

int x=89;

void func2(void)

{

x++;

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

}

10.

#include<stdio.h>

#include<stdlib.h>

int \*func(void);

int main(void)

{

const int \*ptr=func();

\*ptr=7;

printf("\*ptr=%d",\*ptr);

return 0;

}

int \*func(void)

{

int \*p=(int \*)malloc(sizeof(int));

return p;

}

11.

#include<stdio.h>

#include<stdlib.h>

int \*func(void);

int main(void)

{

const int \*ptr=func();

\*ptr=7;

printf("\*ptr=%d",\*ptr);

return 0;

}

int \*func(void)

{

int \*p=(int \*)malloc(sizeof(int));

return p;

}

12.

#include<stdio.h>

int main(void)

{

char str1[]="hockey";

char str2[]="Cricket";

char \*const p=str1;

\*p='j';

p=str2;

return 0;

}

13.

#include<stdio.h>

void func(int a[],const int b[],int c[]);

int main(void)

{

int a[]={1,2,3,4};

int b[]={5,6,7,8};

int c[]={9,10,11,12};

func(a,b,c);

return 0;

}

void func(int a[],const int b[],int c[])

{

a=c;

a[0]++;

b=c;

b[0]++;

}

14.

#include<stdio.h>

int thrice(int i);

int main(void)

{

const int i=23;

const int j=thrice(i);

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

return 0;

}

int thrice(int i)

{

return 3\*i;

}

15.

#include<stdio.h>

int func(void);

int main(void)

{

int i=3;

const int j=i;

const int k=func();

int \*ptr=&k;

const int m=\*ptr;

printf("%d\t%d\t%d\t%d\n",i,j,k,m);

return 0;

}

int func(void)

{

return 4;

}

16.

#include<stdio.h>

#include<stdlib.h>

int \*func(int \*psize);

int main(void)

{

int i,size;

const int \*arr=func(&size);

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

{

printf("Enter a[%d] : ", i);

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

}

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

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

return 0;

}

int \*func(int \*psize)

{

int \*p;

printf("Enter size : ");

scanf("%d",psize);

p =(int \*) malloc(\*psize \* sizeof(int));

return p;

}

17.

#include<stdio.h>

#include<stdarg.h>

void func(int n,...);

int main(void)

{

int count=4;

func(count,2,3,4,5);

return 0;

}

void func(int n,...)

{

va\_list ap;

int a,i;

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

{

a=va\_arg(ap,int);

printf("%d ",a);

}

}

18.

#include<stdio.h>

#include<stdarg.h>

void func(int a, ... ,int b);

int main(void)

{

int a=2,b=3,c=4,d=5;

func(4,2,3,8,5);

return 0;

}

void func(int a, ... ,int b)

{

va\_list \*ap;

va\_start(ap,a);

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

printf("%d",va\_arg(ap,int));

va\_end;

}

19.

#include<stdio.h>

int main(void)

{

int x=6;

++x++;

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

return 0;

}