FUNCTIONS AND RECURSIVE FUNCTIONS

```
1.#include <stdio.h>
int fun(int );
int main() {
int n=5;
printf("%d",fun(n));
}
int fun(int n)
{
   if(n>1)
   return fun(n-2);
   else
   return 2;
}
```

```
2.#include <stdio.h>
void func(int []);
int main() {
int a[]={11,34,67,23,79};
func(a);
}
void func(int b[])
{
   b++;
   printf("%d %d",b[-1],b[2]);
}
```

```
3.#include <stdio.h>
int RecursiveArraySum(int *a,int n)
{
if (n == 0) return 0;
else
return (a[n-1] + RecursiveArraySum(a,n-1));
}
int main( )
{
int a[]={5,10,20,40};
int r=RecursiveArraySum(a,5);
printf("%d",r);
}
```

```
4.#include <stdio.h>
void abc(int );
void xyz(int *);
int main() {
int a=90;
abc(a);
printf("%d",a);
}
```

```
void abc(int a)
{
a=60;
xyz(&a);
printf("%d ",a);
}
void xyz(int *b)
{
*b=70;
}
```

```
5.#include <stdio.h>
void str(char *);
int main() {
char s[]="abcdef";
str(s);
printf("%s",s);
}
void str(char *p)
{
p=p+2;
p[-1]='e';
p[-2]='g';
printf("%s ",p);
}
```

```
6.#include <stdio.h>
int fun(int n);
int main() {
   int n;
   n=fun(6);
   printf("%d",n);
}
int fun(int n)
{
    if(n==3)
    return 4;
    else
    {
        printf("%c",'*');
        return fun(n-1);
    }
}
```

```
7.#include <stdio.h>
char fc(char a);
int main() {
 char a='A';
 printf("%c",fc(a));
}
```

```
char fc(char c)
{
    while(c<'D')
    printf("%c",c++);
    return c++;
}
```

```
8.#include <stdio.h>
int g(int n)
{
if (n < 2)
return n;
return g(n/2);
}
int main ()
{
 int a;
 a=g(14);
 printf("%d", a);
}
```

```
9.#include <stdio.h>
int fun(int);
int main()
{
printf("%d",fun(6));
}
int fun(int n)
{
    if(n>1)
    return fun(--n);
    else
    return 1;
}
```

```
10.#include <stdio.h>
int fun(int);
int main()
{
printf("%d",fun(6));
}
int fun(int n)
{
    if(n>1)
    return fun(n--);
    else
    return 1;
}
```

```
11.#include<stdio.h>
```

```
void string(const char* );
int main()
{
        char s[]="welcome";
        string(s);
        printf("%s",s);
}
void string(const char *p)
{
        while(*p++)
                *p=*p^32;
}
```

```
12.#include<stdio.h>
void string(char*const );
int main()
{
        char s[]="welcome";
        string(s);
        printf("%s",s);
}
void string(char *const p)
{
        while(*p++)
                *p=*p^32;
}
```

```
13.#include<stdio.h>
void val(int,int);
int main()
{
    int a=10;
    val(a,a);
}
void val(int c,int d)
{
        if(c>5)
        {
                printf("#");
        return val(c,c--);
        }
}
```

```
14.#include <stdio.h>
#include <string.h>
void strFunc2 (char A[], int n)
{
 char t;
 if (n <= 1) return;
 t = A[0]; A[0] = A[n-1]; A[n-1] = t;
 strFunc2(&A[1],n-2);
}
int main ()
{
 char A[10] = "PDS 2005";
 strFunc2(A,strlen(A));
 printf("%s", A);
}
```

```
15.#include <stdio.h>
int func ( unsigned int  );
int main() {
          printf("%d", func(4));
}
int func ( unsigned int n )
 {
                  if (n == 0) return 0;
                   return 3*n*(n-1) + func(n-1) + 1;
  }
```

```
16.#include<stdio.h>
void val(int,int);
int main()
{
    int a=10;
    val(a,a);
}
void val(int c,int d)
{
        if(c>5)
        {
                printf("#");
        return val(c--,c);
        }
}
```

```
17.#include<stdio.h>
int mul(int);
int main()
{
    int b;
    b = mul(4);
    printf("%d", b);
}
int mul(int x)
{
    if(x<=1)
     return 1;
    return (x * mul(x-1));
}
```

```
18.#include <stdio.h>
int main()
{
    int num=5;
    int fun(int num);
    printf("%d",fun(num));
}
int fun(int num)
{
    if(num>0)
        return(num+fun(num-2));
return 0;
}
```

```
19.#include <stdio.h>
int main() {
 char r=100;
 int fun(char );
 printf("%d",fun(r));
}
int fun(char s)
{
    if(s)
    fun(++s);
    else
    return s;
}
```

20.What value does the call h(5) return, where h is defined as follows?

```
int h ( int n )
 {
 if (n == 0)
return 1;
 return 2*h(n-1);
}
```

```
21.#include <stdio.h>
int main() {
 char s[]="Born to win";
 void fun(char* );
 fun(s);
}
void fun(char *s)
{
 if(*s!='n')
 {
 fun(++s);
 printf("%c",*s);
 }
 else
 return ;
}
```

```
22.#include <stdio.h>
int main() {
         char s[]="reputation";
          void fun(char* );
           fun(s);
}
void fun(char *s)
{
         if(*s)
                  fun(s+2);
          printf("%c",*s);
}
```

```
23.#include <stdio.h>
int func(int );
int main() {
int n=10;
int r=func(n);
printf("%d",r);
}
int func(int n)
{
    if(n>2)
    return func(n-2)+2;
    else
    return 1;
}
```

24.Write a C program to calculate sum of first 5 natural numbers using recursion.

25.What is the value for the function call S(5,3) for the following code.

```
int S(int n, int k)
 {
if (k > n) return 0;
if ((k == 1) || (k == n)) return 1;
return S(n-1,k-1) + k * S(n-1,k);
}
```

26.How many times ‘*’will be printed in the following code.

```
#include <stdio.h>
int F[10];
int fib(int n)
{
 printf("*");
 if (n <= 1) return 1;
 if (F[n] != 0) return F[n];
 F[n]=(fib(n-1) + fib(n-2));
 return F[n] ;
}
int main( )
{
 int j;
 for (j=0; j< 10; j++) F[j] = 0;
 fib(5);
}
```

```
27.#include<stdio.h>
void fun1(int []);
void main()
{
int a[]={1,2,3,4,6};
fun1(a);
}
void fun1( int a[])
{
int i, sum=0;
for(i=0;i<5;i++)
sum=sum+a[i];
printf("%d",sum);
}
```

```
28.#include <stdio.h>
unsigned int f (unsigned int n)
{
 if (n <= 2) return 1;
 return f(n-3) + f(n-1);
}
int main( )
{
printf("%d",f(10));
}
```

```
29.#include <stdio.h>
int G(unsigned int n)
{
if (n == 0) return 0;
if (n == 1) return 1;
return G(n-2) - G(n-1);
}
int main( )
{
printf("%d",G(6));
}
```

```
30.#include<stdio.h>
```

```
void t(int n, char fp, char tp, char ap)
{
 if(n==1){
 printf("%c%c",fp,tp);
 return;
  }
 t(n-1,fp,ap,tp);
 printf("%c%c",fp,tp);
 t(n-1,ap,tp,fp); return;
}
void main()
{
 t(2,'x','y','z');
}
```

```
31.#include<stdio.h>
 int f(int n)
{
 static int i = 1;
 if (n >= 5)
 return n;
 n = n+i;
 i++;
 return f(n);
}
void main()
{
 printf("%d",f(1));
}
```

```
32.#include<stdio.h>
 void foo(int n, int sum)
{
 int k = 0, j = 0;
 if (n == 0) return;
 k = n % 10;
 j = n / 10;
 sum = sum + k;
 foo (j, sum);
 printf ("%d ", k);
}
int main ()
{
 int a = 526, sum = 0;
 foo (a, sum);
}
```

```
33.#include <stdio.h>
int f(int n)
{
 static int r = 0;
 if (n <= 0) return 1;
 if (n > 3)
 {
 r = n;
 return f(n-2)+2;
 }
 return f(n-1)+r;
}
int main()
{
 printf("%d", f(5));
}
```

```
34.#include<stdio.h>
unsigned int fun(unsigned int n, unsigned int r)
{
 if (n > 0) return (n%r + fun (n/r, r ));
 else return 0;
}
int main()
{
    int r=fun(345,10);
    printf("%d",r);
}
```

35.Consider the following C function.

```
int fun (int n)
 {
 int x=1, k;
 if (n==1)
 return x;
 for (k=1; k<n; ++k)
x = x + fun(k) * fun(n – k);
 return x;
}
```

The return value of fun(5) is ___________.

36.Consider the following recursive C function. If get(6) function is being called in main() ,then how many times will the get() function be invoked before returning to the main()?

```
void get (int n)
 {
 if (n < 1) return;
 get(n-1);
 get(n-3);
 printf("%d", n);
 }
```

```
37.#include <stdio.h>
int main() {
   int i=3;
  if(i)
  {
      printf("%d ",i);
      i--;
      main();
  }
  return 0;
}
```

```
38.#include <stdio.h>
int main() {
  static int i=3;
  if(i)
  {
      printf("%d ",i);
      i--;
      main();
  }
  return 0;
}
```

```
39.#include<stdio.h>
int abc( unsigned int a , unsigned int b )
 {
 if (b == 0) return a;
 return abc(b,a%b);
 }
int main()
{
    printf("%d",abc(-3,-4));
}
```

```
40.#include<stdio.h>
int fun ( unsigned int a , unsigned int b )
         {
                  if ((a == 0) || (b == 0)) return 0;
                   return a * b / fun(a,b);
                     }
int main()
{
            printf("%d",fun(-4,-2));
}
```

```
41.#include<stdio.h>
 int str ( char A[] , unsigned int n )
 {
 int t;
 if (n == 0) return 0;
 t = str(A,n-1);
 if ( ((A[n-1]>='a') && (A[n-1]<='z')) || ((A[n-1]>='A') && (A[n-1]<='Z')) )
 ++t;
 return t;
 }
int main()
{
   printf("%d",str("AhK#*L()",8));
}
```

42.Consider the following function

```
int f(int j)
{
 static int i=50;
 int k;
 if(i==j)
 {
 printf("something");
 k=f(i);
 return 0;
 }
 else
 return 0;
}
```

a) Function will return 0 for all j
b) Function prints "something " for all values of j
c) For i==j function goes into infinite loop
d) For i==j function returns 0

```
43.#include<stdio.h>
double f(double x){
 if ((x*x - 3) < 0.01) return x;
 else return f(x/2);
}
int main()
{
  printf("%f",f(3)) ;
}
```

44. Write a C program to find maximum and minimum elements in array using recursion.

45. Write a C program to find sum of elements of array using recursion.

46. Write a C program to find power of any number using recursion.

47.What value will the following function return when called as recur(3)?

```
 int recur(int data)
 {
 int k;
  k=(data>2)?(recur(data-1)-recur(data-2)):1;
 return k;
 }
```

```
48.#include <stdio.h>
unsigned int f(unsigned int n)
{
 if (n < 10)
printf("%d",n);
 else {
 printf(" %d ", n%10);
 f(n/10);
 printf(" %d ", n%10);
 }
return 1;
}
int main ()
{
 printf("%d", f(15));
}
```

```
49.#include<stdio.h>
int fun ( unsigned int n )
 {
 if (n == 0) return 0;
 if (n & 1) return -1;
 return 1 + fun(n-1);
 }
int main()
{
    int x;
    x=fun(-6);
    printf("%d",x);
}
```

```
50.#include<stdio.h>
int fn ( int n )
 {
 if (!(n & 1)) return -2;
 return 2 + fn(n-1);
 }
int main()
{
    int x;
    x=fn(7);
    printf("%d",x);
}
```