

ASSIGNMENT-9

Write a C Program for the following problem statements:-

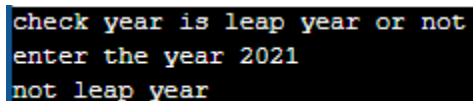
a) Functions without arguments and without return type

- Check whether the year is Leap year

```
#include <stdio.h>
void check();
int main()
{

    printf("check year is leap year or not\n");
    check();
}
void check()
{
    int year;
    printf("enter the year ");
    scanf("%d",&year);
    if(year%4==0 | year%400==0)
    {
        printf("leap year");
    }
    else
    {
        printf("not leap year");
    }
}
```

Output:-



```
check year is leap year or not
enter the year 2021
not leap year
```

- Count number of digits in a number

```
#include <stdio.h>
void check();
int main()
{

    printf("count number of digits in a number\n");
```

```

    check();
}
void check()
{
    int n,count=0;
    printf("enter the number ");
    scanf("%d",&n);
    while(n!=0)
    {
        n=n/10;
        ++count;
    }
    printf("%d number of digits in a given number",count);
}

```

Output:-

```

count number of digits in a number
enter the number 2435
4 number of digits in a given number

```

b) Functions without arguments and with return type

```

#include <stdio.h>
int calc(int);

int main()

{

    int f,c;

    printf("convert the temperature from fahrenheit to celsius: ");
    scanf("%d",&f);

    c=(f - 32) * 5 / 9;

    printf("degree in celsius: %d",c);

}

int calc(int f)

```

```

{

return (f - 32) * 5 / 9;

}

```

Output:-

```

convert the temperature from fahrenheit to celsius: 20
degree in celsius: -6

```

- Check Armstrong number or not

```

#include <stdio.h>
int ams( );
void main( )
{
    int amst;
    printf("checking amstrong or not.\n");
    amst=ams( );
    if(amst==0)
        printf("your no is amstrong.");
    else
        printf("your no is not amstrong.");
}
int ams( )
{
    int no,rem,sum=0,a;
    printf("enter a no : ");
    scanf ("%d",&no);
    a=no;
    while(no>0)
    {
        rem=no%10;
        sum=sum+(rem*rem*rem);
        no=no/10;
    }
    if (sum==a)
    {
        return 0;
    }
}

```

Output:-

```

checking amstrong or not.
enter a no : 153
your no is amstrong.

```

c) Functions with arguments and without return type
- check prime number or not

```
#include<stdio.h>
int CheckPrime();
int main()
{
    int n1,prime;
    printf(" enter a number : ");
    scanf("%d",&n1);
    prime = CheckPrime(n1);
    if(prime==1)
        printf(" number is a prime number.\n");
    else
        printf(" The number is not a prime number.\n");
    return 0;
}
int CheckPrime(n1)
{
    int i=2;
    while(i<=n1/2)
    {
        if(n1%i==0)
            return 0;
        else
            i++;
    }
    return 1;
}
```

Output:-

```
enter a number : 5
number is a prime number.
```

- Find ASCII number to character and character to ASCII number

```
#include <stdio.h>
int asc(int,char);
int main( )
{
    char c;
```

```

int n;
printf("Enter a character: ");
scanf("%c", &c);
printf("enter a number: ");
scanf("%d",&n);
asc(n,c);
}
int asc(int n,char c)
{

    printf("ASCII value of character \"%c\" = %d", c, c);
    printf("\nASCII value of number of character \"%d\" = %c", n, n);

    return 0;
}

```

Output:-

```

Enter a character: C
enter a number: 65
ASCII value of character "C" = 67
ASCII value of number of character "65" = A

```

d) Functions with arguments and with return type

- Calculate factorial of a number

```

#include <stdio.h>
int factorial(int num)
{
    int i,fact=1;

    for (i = 1; i <= num; i++)
        fact = fact * i;
    return fact;
}

int main()
{
    int num;
    printf("Enter a number to find the factorial ");
    scanf("%d", &num);
    if(num<0)
    {
        printf("Factorial is not defined for negative numbers.");
    }
    printf("Factorial of %d = %d", num, factorial(num));
}

```

```
    return 0;  
}
```

Output:-

```
Enter a number to find the factorial 5  
Factorial of 5 = 120
```

- Count number of digits in a number

```
#include <stdio.h>  
int countDigit(int n)  
{  
    int count = 0;  
    while (n != 0)  
    {  
        n = n / 10;  
        ++count;  
    }  
    return count;  
}  
  
int main()  
{  
    int n;  
    printf("enter the number ");  
    scanf("%d",&n);  
    printf("Number of digits : %d", countDigit(n));  
    return 0;  
}
```

Output:-

```
enter the number 3456  
Number of digits : 4
```

e) Function return Multiple values

- Largest and smallest of five numbers

```
#include <stdio.h>
int ls(int [],int *);
int main( )
{
    int no[5],i, rp,f[2];
    printf("Enter five different nos: ");
    for(i=0;i<5;i++)
    {
        scanf("%d",&no[i]);
    }
    rp=ls(no,f);
    printf("largest no is: \"%d\" and smallest is : \"%d\"",f[0],f[1]);
    return 0;
}
int ls(int a[],int *p)
{
    int i,sm,lg;
    lg=sm=a[i];
    for(i=0;i<5;i++)
    {
        if (a[i]>lg)
            lg=a[i];
        else if (a[i]<sm)
            sm=a[i];
    }
    *(p+0)=lg;
    *(p+1)=sm;
    return p;
}
```

Output:-

```
Enter five different nos: 3 45 2 3 5
largest no is: "45" and smallest is : "2"
```

- Find Simple interest and compound interest

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

```
#include<math.h>
```

```

int interest(float,float,float ,float*);

int main() {

    float amt,t,r,g[2];

    printf("Enter Principal Amount: ");

    scanf("%f",&amt);

    printf("Enter Time Period in Year: ");

    scanf("%f",&t);

    printf("Enter Rate of Interest Per Year: ");

    scanf("%f",&r);

    interest(amt,t,r,g);

    printf("compound interest is: \" %f \" and simple interest is : \" %f \"",g[1],g[0]);

    return 0;

}

int interest(float p,float r,float t ,float*i)

{

    float si,ci;

    si=(p*t*r)/100;

    *(i+0)=si;

    ci= p*(pow((1+r/100),t));

    *(i+1)=ci;

    return *(i+0),*(i+1);

}

```

Output:-

```

Enter Principal Amount: 234
Enter Time Period in Year: 3
Enter Rate of Interest Per Year: 21
compound interest is: " 435.308685 " and simple interest is : " 147.419998 "

```


f) Nesting of Functions

- Print the sum of series $1 + 1/2 + 1/3 + 1/4 + \dots + 1/N$.

```
#include <stdio.h>
int main()
{
    float a;
    float input()
    {
        float z,c;
        printf("enter the value of N : ");
        scanf("%f",&c);
        float proc(float n)
        {
            float i,b=0;
            for (i=1;i<n;i++)
            {
                b=b+1/i;
            }
            printf(" the value of the series is : %f",b);
        }
        proc(c);
    }
    input();
    return 0;
}
```

Output:-

```
enter the value of N : 8
the value of the series is : 2.592857
```

- reverse a number

```
#include <stdio.h>

int main()
{ int a;
  auto int input(){
    int z,c;
    printf("enter the value of N :");
    scanf("%d",&c);
```

```

int proc(int n){
int o,b=0;
for (o=1;n!=0;o++){
    b=b+n%10;
    b=b*10;
    n=n/10;
}b=b/10;
printf(" the reverse of N is : %d",b);
}
proc(c);
}
input();
return 0;
}

```

Output:-

```

enter the value of N :456
the reverse of N is : 654

```

g) Recursive Functions

- to Print Fibonacci Series

```

#include <stdio.h>
int fib(int);
void main() {
    int no,f;
    printf("enter a no ");
    scanf("%d",&no);
    f=fib(no);
    printf("fibonannci value is : %d",f);
}
int fib(int n)
{
    if(n==0)
        return 0;
    else if (n==1)
        return 1;
    else
        return fib(n-1)+fib(n-2);
}

```

```
}
```

Output:-

```
enter a no 7
fibonannci value is : 13
```

- to convert a decimal number to binary

```
#include <stdio.h>
int convert(int);
int main()
{
    int dm,b;
    printf("enter a decimal no: ");
    scanf("%d",&dm);
    b=convert(dm);
    printf("binary value is : %d", b);
    return 0;
}
int convert(int no)
{
    if (no == 0)
        return 0;
    else
        return (no % 2 + 10 * convert(no/2));
}
```

Output:-

```
enter a decimal no: 432
binary value is : 110110000
```

h) Passing 1D Array in Functions

- Reverse the elements of an array

```
#include<stdio.h>
int main()
{
    int arr[50],size,i;
    printf("Enter size : ");
    scanf("%d",&size);
    printf("Enter %d numbers : ",size);
```

```

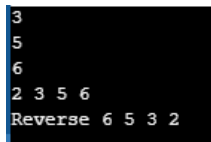
    for(i=0; i<size; i++)
        scanf("%d",&arr[i]);
    for(i=0; i<size; i++)
        printf("%d ",arr[i]);
    printf("\nReverse ");
    rev(arr,size);
    for(i=0; i<size; i++)
        printf("%d ",arr[i]);

    return 0;
}

int rev(int arr[],int size)
{
    int i,temp;
    for(i=0; i<size/2; i++)
    {
        temp=arr[i];
        arr[i]=arr[size-i-1];
        arr[size-i-1]=temp;
    }
}

```

Output:-



```

3
5
6
2 3 5 6
Reverse 6 5 3 2

```

- Find the fourth largest and Third smallest element in an array

```

#include<stdio.h>
int main()
{
    int arr[50],size,i;
    printf("Enter size : ");
    scanf("%d",&size);
    printf("Enter %d numbers : ",size);
    for(i=0; i<size; i++)
        scanf("%d",&arr[i]);
    largest(arr,size);
    return 0;
}

```

```

void largest(int arr[],int size)
{
int large1,large2,large3,large4,i;
large1=large2=large3=large4=arr[0];
for(i=0; i<size; i++)
{
if(arr[i]>large1)
{
large4=large3;
large3=large2;
large2=large1;
large1=arr[i];
}
else if(arr[i]>large2 && large1!=arr[i])
{
large4=large3;
large3=large2;
large2=arr[i];
}
else if(arr[i]>large3 && large2!=arr[i])
{
large4=large3;
large3=arr[i];
}
else if(arr[i]>large4 && large3!=arr[i])
{
large4=arr[i];
}
}
printf("3rd largest %d\n4th largest %d",large3,large4);
}

```

Output:-

```

56
90
34
3rd largest 34
4th largest 12

```

i) Passing 2D Array in Functions

- Sum of upper triangular and lower triangular elements of mxm array

```

#include <stdio.h>
void sum(int mat[3][3], int r, int c)
{
    int i, j;
    int usum = 0;
    int lsum = 0;
    for (i = 0; i < r; i++)
        for (j = 0; j < c; j++)
        {
            if (i <= j)
            {
                usum += mat[i][j];
            }
        }
    printf("Upper sum is %d\n", usum);
    for (i = 0; i < r; i++)
        for (j = 0; j < c; j++)
        {
            if (j <= i)
            {
                lsum += mat[i][j];
            }
        }
    printf("Lower sum is %d", lsum);
}
int main()
{
    int r = 3;
    int c = 3;
    int mat[3][3] = {{ 6, 5, 4 },
                     { 1, 2, 5 },
                     { 7, 9, 7 }};
    sum(mat, r, c);
    return 0;
}

```

Output:-

```

Upper sum is 29
Lower sum is 32

```

- Perform matrix multiplication between two mxn array

```
#include <stdio.h>
#define N 4
void multiply(int mat1[][N], int mat2[][N], int res[][N])
{
    int i, j, k;
    for (i = 0; i < N; i++)
    {
        for (j = 0; j < N; j++)
        {
            res[i][j] = 0;
            for (k = 0; k < N; k++)
                res[i][j] += mat1[i][k] * mat2[k][j];
        }
    }
}
int main()
{
    int mat1[N][N] = { { 1, 1, 1, 1 },
                        { 2, 2, 2, 2 },
                        { 3, 3, 3, 3 },
                        { 4, 4, 4, 4 } };
    int mat2[N][N] = { { 1, 1, 1, 1 },
                        { 2, 2, 2, 2 },
                        { 3, 3, 3, 3 },
                        { 4, 4, 4, 4 } };
    int res[N][N];
    int i, j;
    multiply(mat1, mat2, res);
    printf("Result matrix is \n");
    for (i = 0; i < N; i++)
    {
        for (j = 0; j < N; j++)
            printf("%d ", res[i][j]);
        printf("\n");
    }
    return 0;
}
```

Output:-

```
10 10 10 10
20 20 20 20
30 30 30 30
40 40 40 40
```

j) Passing Strings in Functions

- to perform Substring Extraction (With and Without String Handling Functions).

```
#include <stdio.h>
int show(char[]);
int main()
{
    char a[20]="he is a good boy.";
    show(a);
    return 0;
}
int show(char s[])
{
    char *b;
    printf("substring is : %s",b=strstr(s,"a"));
}
```

Output:-

```
substring is : a good boy.
```

- to read a string and prints if it is a palindrome or not.

```
#include<stdio.h>
#include<string.h>
int pal(char[]);
int main()
{
    char str[20];
    char c;

    printf("Enter a string: ");
    scanf("%s", str);
    c=pal(str);
    if (c==1)
    {
        printf("%s is not a palindrome", str);
    }
}
```



```
else
{
printf("%s is a palindrome", str);
}
}
```

```
int pal(char s[])
{
int i, len,c;
c= 0;
len = strlen(s);
for(i=0;i < len ;i++)
{
if(s[i] != s[len-i-1])
{
c=1;
break;
}
}
return c;
}
```

Output:-

```
Enter a string: malayalam
malayalam is a palindrome
```