

ASSIGNMENT 9 ANSWER

NAME –ABHISEK DAS

REGD NO- 2061020020

a) Functions without arguments and without return type
i. check whether the year is Leap year

program:-

```
#include <stdio.h>

void lp();

int main()
{
    printf("checking leap year.\n");
    lp();
    return 0;
}

void lp()
{
    int y;
    printf("enter year no: \n");
    scanf("%d",&y);
    if(y%4==0)
        printf("your year \"%d\" is leap year.\n",y);
    else
        printf("your year \"%d\" is not a leap year.\n",y);
}
```

```
}
```

Output:-

checking leap year

enter year no:

2001

your year "2001" is not a leap year.

ii. convert binary to hexadecimal

Program:-

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

```
void hexa();
```

```
int main()
```

```
{
```

```
    printf("conversion of binary to hexadecimal.\n");
```

```
    hexa();
```

```
    return 0;
```

```
}
```

```
void hexa()
```

```
{
```

```
long int b, hd = 0, i = 1, rem;
```

```

printf("Enter the binary number: ");
scanf("%ld", &b);
while(b!=0)
{
    rem = b % 10;
    hd = hd + rem * i;
    i = i * 2;
    b = b / 10;
}
printf("Equivalent hexadecimal value: %lX", hd);
}

```

Output:-

conversion of binary to hexadecimal.

Enter the binary number: 1111

Equivalent hexadecimal value: F

iii. count number of digits in a number

Program:-

```

#include <stdio.h>
void count();
int main()
{
    printf("counting the no of digits.\n");
    count();
    return 0;
}
void count()

```

```

{
    int n; // variable declaration
    int count=0; // variable declaration
    printf("Enter a number: ");
    scanf("%d",&n);
    while(n!=0)
    {
        n=n/10;
        count++;
    }
    printf("\nThe number of digits in the above no is : %d",count);
}

```

Output:-

counting the no of digits.

Enter a number: 20

The number of digits in the above no is : 2

b) Functions without arguments and with return type

i.check Armstrong number or not

program:-

```

#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 : 154
your no is not amstrong.

```

ii.To evaluate the following using loops $x + x^3 / 3! + x^5 / 5! + \dots$ upto 5 terms

program:-

```

#include <stdio.h>
int loop();
void main()
{
    int sum;
    printf("evaluate the following using loops  $x + x^3 / 3! + x^5 / 5! + \dots$ \n");
    sum=loop();

    printf("sum of term is : %d ",sum);
}
int loop()
{
    int x,sum,inc,i;
    int n,p,nn,m,d;
    printf("Enter value of x: ");
    scanf("%d",&x);
    printf("enter the no of terms: ");
    scanf("%d",&n);
    sum=x;
    i=1;
    m=-1;
    printf("The value of the series:\n ");
    printf("%d",x);
    do{
        //incrementing the power
        inc=(2*i+1);
        //power calculation
        p= pow(x,inc);
    }
}

```

```

//multiply into m
nn=p*m;
//printing the new nos
printf("\n%d\n",nn);
sum=sum+nn;
m=m*(-1);
i++;
}while(i<n);
    return sum;
}

```

output:-

evaluate the following using loops $x + x^3 / 3! + x^5 / 5! + ..$

Enter value of x: 2

enter the no of terms: 5

The value of the series:

2

-8

32

-128

512

sum of term is : 410

iii.Convert temperature Fahrenheit to Celsius program :-

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

```
int fc();
```

```
int main()
```

```
{
```

```
    float c;
```

```
    printf("converting farenhite to celcious...\n ");
```

```
    c=fc();
```

```
    printf("celcious is: %f ",c);
```

```

}

int fc()

{
    float fa,cel;

    printf("enter temperture in farenhite: ");

    scanf("%f",&fa);

    cel= (fa-32)* 5/9;

    return cel;
}

```

output:-

converting farenhite to celcious...
 enter temperture in farenhite: 72
 celcious is: 22.000000

c) Functions with arguments and without return type

i. check prime number or not

program:-

```

#include<stdio.h>
int prim(int);
int main()
{
    int no;
    printf("checking prime or not...\n ");
    printf("enter your no: ");
    scanf("%d",&no);
    prim(no);
}
int prim(int n)
{
    int i,c=0;
    for(i=2;i<=n/2;i++)
    {
        if (n%i==0)
        {
            c=1;
        }
    }
}

```

```

if (n==0)
printf("your no is zero.");
else if (n==1)
printf("your no natural no.");
else if (c==0)
printf("your no %d is prime",n);
else
printf("your no %d is not prime.",n);
}

```

output:-

```

checking prime or not...
enter your no: 3
your no 3 is prime

```

**ii. find all roots of the quadratic equation
program:-**

```

#include<stdio.h>
#include<math.h>
int quad(float,float,float);
int main ()
{
    float a,b,c;
    printf("Enter the value of a : ");
    scanf("%f",&a);
    printf("Enter the value of b : ");
    scanf("%f",&b);
    printf("Enter the value of c : ");
    scanf("%f",&c);
    quad(a,b,c);
}
int quad(float a,float b,float c)
{
    float r1,r2,d,img;
    d=(b*b)-4*a*c;
    if (d>0)
    {

```



```

r1= (-b+ sqrt (d))/ 2*a ;
r2=(-b- sqrt (d))/ 2*a ;
printf("Quadratic equation has two roots \n");
printf("two roots are r1=%g & r2= %g ",r1,r2);
}
else if(d==0)
{
    r1=r2=(-b/2*a);
    printf("two roots are equal\n");
    printf("r1= %g & r2= %g",r1,r2);
}
else if (d<0)
{
    r1=r2=(-b/2*a);
    img= sqrt(-d)/2*a;
    printf("two roots are imaginary= %g \n",img);
}
return 0;
}

```

output:-

```

Enter the value of a : 1
Enter the value of b : 2
Enter the value of c : -6
Quadratic equation has two roots
two roots are r1=1.64575 & r2= -3.64575

```

iii. find ASCII number to character and character to ASCII number

program:-

```

#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: D
enter a number: 66
ASCII value of character "D" = 68
ASCII value of number of character "66" = B

```

d) Functions with arguments and with return type

i) check perfect or abundant or deficient number

program:-

```

#include <stdio.h>

int pn(int);

int main()
{
    int no,p=0;

    printf("enter a no : \n");

    scanf("%d",&no);

    p = pn(no);

    if (p==0)

```

```
    printf("no %d is a perfect no.",no);  
    else  
        printf("no %d is not a perfect no.",no);  
    return 0;  
}  
  
int pn(int a)  
{  
    int i,sum=0;  
    for(i=1;i<=a/2;i++)  
    {  
        if (a%i==0)  
        {  
            sum=sum+i;  
        }  
    }  
    if (sum==a)  
        return 0;  
    else  
        return 1;  
}
```

Output:-

enter a no :

28

no 28 is a perfect no.

ii) calculate factorial of a number

program:-

```
#include <stdio.h>
int fact(int);
int main() {
    int no,f;
    printf("Enter the no : ");
    scanf("%d", &no);
    f=fact(no);
    printf("factorial of \"%d\" is = %d ",no,f);
}
int fact(int n)
{
    int i,f=1;
    if (n==0)
    {
        return 0;
    }
    else if (n==1)
    {
        return 1;
    }
    else
    {
        for(i=1;i<=n;i++)
        {
            f=f*i;
        }
        return f;
    }
}
```

```
}
```

```
}
```

output:-

Enter the no : 6

factorial of "6" is = 720

iii) count number of digits in a number

program:-

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

```
int count(int);
```

```
int main()
```

```
{
```

```
    int n,nn;
```

```
    printf("counting the no of digits.\n");
```

```
    printf("Enter a number: ");
```

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

```
    nn=count(n);
```

```
    printf("\nThe number of digits in the above no is : %d",nn);
```

```
    return 0;
```

```
}
```

```
int count(int no)
```

```
{
```

```
    int c=0;
```

```
    while(no!=0)
```

```
    {
```

```
        no=no/10;
```

```
        c++;
```

```
    }
```

```
    return c;
```

```
}
```

output:-

counting the no of digits.

Enter a number: 300

The number of digits in the above no is : 3

**e) Function return Multiple values
Largest and Smallest of five numbers**

Program:-

```
#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[0];
    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: 70
80
90
45
65
largest no is: "90" and smallest is : "45"

Find Simple interest and compound interest**Program:-**

```
#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: 10000

Enter Time Period in Year: 12

Enter Rate of Interest Per Year: 10

compound interest is: " 31058.484375 " and simple interest is : " 12000.000000 "

simple calculator (add, sub, mul, div, mod)

Program:-

```
#include <stdio.h>
int calc(int,int,int*,int*,int*,int*,int*);
int main()
{
    int no1,no2,a,s,m,d,mo;
    printf("enter the value of 1st no: ");
    scanf("%d",&no1);
    printf("enter the value of 2nd no: ");
    scanf("%d",&no2);

    calc(no1,no2,&a,&s,&m,&d,&mo);
    printf("\n after addition : %d \n after subtraction : %d \n after multiplication: %d",a,s,m);
    printf("\n after division: %d \n the module is : %d",d,mo);
    return 0;
}
int calc(int x,int z,int* a,int* s, int* m,int* d,int* mo)
{
    *a=x+z;
    *s=x-z;
    *m=x*z;
    *d=x/z;
    *mo=x%z;
}
```

Output:-

enter the value of 1st no: 6
enter the value of 2nd no: 4

after addition : 10
after subtraction : 2
after multiplication: 24
after division: 1
the module is : 2

f) Nesting of Functions

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

Program:-

```
#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 : 6
the value of the series is : 2.283334

reverse a number

Program:-

```
#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 :115
the reverse of N is : 511
```

g) Recursive Functions to Print Fibonacci Series

Program:-

```
#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 12

fibonannci value is : 144

to convert a decimal number to binary

Program:-

```
#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: 10
binary value is : 1010

h) Passing 1D Array in Functions

Reverse the elements of an array

Program:-

```
#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

```

Enter size : 3
Enter 3 numbers : 1
2
5
1 2 5
Reverse 5 2 1

```

Find the fourth largest and Third smallest element in an array

Program:-

```

#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

```

Enter size : 5
Enter 5 numbers : 4
5
6
7
8
3rd largest 6
4th largest 5

```

i) Passing 2D Array in Functions

Sum of upper triangular and lower triangular elements of mxm array

Program:-

```
#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

Program:-

```

#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

```

Result matrix is
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).
Program:-**

```
#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.

Program:- without string func

```
#include <stdio.h>
int show(char[],int );
int main()
{
    char str[100] ;
    int pos, len;
    printf("Input a string: ");
    gets(str);
    printf("Enter the starting position of substring: ");
    scanf("%d", &pos);
```

```

show(str,pos);
return 0;
}

int show(char a[],int p)
{
    char sub[100];
    int i=0,l;
    l=strlen(a);
    while (i<l)
    {
        sub[i]=a[p+i-1];
        i++;
    }
    sub[i]='\0';
    printf("substring is : %s",sub);
}

```

Output

Input a string: he is agood boy
Enter the starting position of substring: 2
substring is : e is agood boy

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

Program:-

```

#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: guug  
guug is a palindrome
```