Programs for External/Practical exam from Unit 1 to 4

- Write a program to find the addition, subtraction, multiplication, division if two numbers. (Pg. 25/26).

SOLUTION:

```
#include<stdio.h>
#include<conio.h>
// Write a program to find the addition, subtraction, multiplication and division of two numbers.
int main()
{
	int r1,r2,r3,r4,num1,num2;
	num1=11,num2=9;
	printf("\n num1=%d num2=%d",num1,num2);
	r1=num1+num2;
	printf("\n num1+num2=%d",r1);
	r2=num1-num2;
	printf("\n num1-num2=%d",r2);
	r3=num1*num2;
	printf("\n num1*num2=%d",r3);
	r4=num1/num2;
	printf("\n num1/num2=%d",r4);
	getch();
	return 0;
}
```

- Program of #define preprocessor directive PI. (pg. 39).

SOLUTION:

```
#include<stdio.h>
#include<conio.h>
#define PI 3.14
//program of #define  preprocessor directive PI
int main()
{
	int radius;
	float area;
	printf("Enter radius=");
	scanf("%d",&radius);
	area = PI*radius*radius;
	printf("\nArea of circle:%2f",area);
	getch();
	return 0;
}
```

- Program of #define preprocessor directive array size. (pg. 39).

SOLUTION:

```
#include<stdio.h>
#include<conio.h>
#define SIZE 5
// Program of #define preprocessor directive array SIZE
int main()
{
	int num[SIZE],i;
	for(i=0;i<SIZE;i++)
	{
		printf("Enter any number:");
		scanf("\n%d", &num[i]);
	}
	printf("\nArray elements are:\n");
	for(i=0;i<SIZE;i++)
	{
		printf("%d\t", num[i]);
	}
	getch();
	return 0;
}
```

- The #define preprocessor as functions. (pg. 40).

SOLUTION:

```
#include<stdio.h>
#include<conio.h>
#define SQUARE(x) x*x
//program of #define preprocessor as function
int main()
{
	int num;
	printf("Enter any Number:");
	scanf("%d", &num);
	printf("\n The Square is :%d",SQUARE(num));
	getch();
	return 0;
}
```

- Program of #if-else-#endif Preprocessor directive.(pg. 41).

SOLUTION:

```
#include<stdio.h>
#include<conio.h>
#define MAX 50 //program of #if-#else-#endif preprocessor directive
int main(){
	#if MAX>20
	printf("Yes, MAX is greater than 20.");
	#else
	printf("No,MAX is not greater than 20.");
	#endif
	getch();	return 0;	}
```

- Check whether the given number is greater than 5 or not. (Pg. 44).

SOLUTION:

```
#include<stdio.h>
#include<conio.h>
// Check wether the given number is greater than 5 or not.
int main()
{
	int num;
	//Initialize and read in a value for num1
	printf("\nEnter an integer between 1 and 10:");
	scanf("%d",&num);
	if(num>5)
	{
		printf("You entered %d which is greater than 5\n", num);
	}
	getch();
	return 0;
}
```

- Program to check whether the number is even or odd. (pg. 45).

SOLUTION:

```
#include<stdio.h>
#include<conio.h>
//program to check whether the number is even or odd
int main()
{
	int num1;
	//Initialize and read in a value for num1
	printf("\n Enter any Number:");
	scanf("%d",&num1);
	if((num1%2)==0)//checking condition for even or odd
	{
		printf("\n %d Number is Even.",num1);
	}
	else
	{
		printf("\n%d Number is Odd.",num1);
	}
	getch();
	return 0;
}
```

- Program to check whether the number is positive, negative or zero. (pg. 46).

SOLUTION:

```
#include<stdio.h>
#include<conio.h>
//program to check wether the number is positive ,negative or zero.
int main()
{
	int num1;
	//initialize and read in a value for num1.
	printf("\nEnter any number:");
	scanf("%d",&num1);
	if(num1>0)
	{
		printf("\n%d Number is positive.",num1);
	}
	else if(num1<0)
	{
		printf("\n%d Number is negative.",num1);
	}
	else
	{
		printf("\n%d Number is zero.",num1);
	}
	getch();
	return 0;
}
```

- Program to enter a number from the user and display the month name. if number>13 then display “invalid input” using switch case. (Pg. 48).

SOLUTION:

```
#include<stdio.h>
#include<conio.h>
//Program to enter a number from user and display the month name,if number>13 then display invalid input using switch case
int main()
{
	int num1;
	//initialize and read in a value for num1.
	printf("\nEnter month number:");
	scanf("\n%d", &num1);
	switch(num1)
	{
		case 1:printf("January.");
			break;
		case 2:printf("February.");
			break;
		case 3:printf("March.");
			break;
		case 4:printf("April.");
			break;
		case 5:printf("May.");
			break;
		case 6:printf("June.");
```

```
                break;
        case 7:printf("July.");
                break;
        case 8:printf("August.");
                break;
        case 9:printf("September.");
                break;
        case 10:printf("October.");
                break;
        case 11:printf("November.");
                break;
        case 12:printf("December.");
                break;
        default:printf("INVALID INPUT.");
    }
    getch();
    return 0;
}
```

- Program to print 1 to 10 numbers using while loop.(Pg.50)

SOLUTION:

```
#include<stdio.h>
#include<conio.h>
// program to print 1 to 10 numbers.
int main()
{
	int num1;
	num1=1;
	while(num1<=10)
	{
		printf("\t%d", num1);
		num1++;
	}
	getch();
	return 0;
}
```

- Program to use do-while loop. (Pg.50/51).

SOLUTION:

```
#include<stdio.h>
#include<conio.h>
//program to print 1-10 using do-whule loop.
int main()
{
	int num1;
	num1=1;
	do
	{
		printf("\t%d",num1);
		num1++;
	}while(num1<=10);
	getch();
	return 0;
}
```

- Program to print 1 to 10 number using for loop.(pg. 51).

SOLUTION:

```
#include<stdio.h>
#include<conio.h>
//program to print 1 to 10 numbers using for loop.
int main()
{
	int num1;
	for (num1=1;num1<=10;num1++)
	{
		printf("\t%d",num1);
	}
	getch();
	return 0;
}
```

- Program to find the factorial of a number using for loop. (Pg. 52).

SOLUTION:

```
#include<stdio.h>
#include<conio.h>	//program to find the factorial of a number.
int main(){
	int num,fact,i;
	fact=1;
	printf("\nEnter any Number:");
	scanf("%d",&num);	//calculating the factorial
	for(i=1;i<=num;i++)	{
		fact=fact*i;	}
	printf("Factorial of %d = %d",num,fact);	return 0;	}
```

- Program to find the largest of three numbers using if-else. (Pg.53).

SOLUTION:

```
#include<stdio.h>
#include<conio.h>
//program to find the largest of three numbers.
int main()
{
	int num1,num2,num3;
	// initialize and read the three numbers for num1,num2,num3.
	printf("\nEnter any three numbers:");
	scanf("%d %d %d",&num1,&num2,&num3);
	if(num1>num2&&num1>num3)
	{
		printf("\n%d Number is greater number.",num1);
	}
	else if(num2>num1&&num2>num3)
	{
		printf("\n%d Number is greater number.",num2);
	}
	else
	{
		printf("\n%d Number is greater number.",num3);
	}
	return 0;
}
```

- Write a program to find the sum of squares of digits of numbers (Pg. 54).

SOLUTION:

```
#include<stdio.h>
#include<conio.h>
//Write a program to find the sum of squares of digits of numbers
int main()
{
	int num,i,sum=0;
	// initialize and read in a value for num.
	printf("\nEnter Number:");
	scanf("%d",&num);
	//calculating the sum square of digit
	for(i=1;i<=num;i++)
	{
		sum=sum+(i*i);
	}
	printf("\nSum of square of digits = %d",sum);
	return 0;
}
```

- Write a program to print the Fibonacci series (Pg. 55).

SOLUTION:

```
#include<stdio.h>
#include<conio.h>
//write a program to print the fibonacci series.
int main()
{
	int i,a,b,c,num;
	a=0;
	b=1;
	//initialize and read in a value for num
	printf("\nEnter number:");
	scanf("%d",&num);
	printf("\nFibonacci series up to %d term \n",num);
	// by default fibonaaci series starting values 0 and 1.
	printf("%d\t%d",a,b);
	// Remaining fibonacci series starting values calculating.
	for(i=3;i<=num;i++)
	{
		c=a+b;
		printf("\t%d",c);
		a=b;
		b=c;
	}
	return 0;
}
```

- Write a program that solves Quadratic equation (Pg. 57/58).

SOLUTION:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
//Write a program that solves Quadratic equation
int main()
{
	float a, b, c, x1, x2, determinant, realpart, imaginarypart;
	printf("Enter coefficients a, b and c:");
	scanf("%f %f %f",&a, &b, &c);
	determinant = b*b-4*a*c;
	if(determinant>0)
	{
		x1=(-b+sqrt(determinant))/(2*a);
		x2=(-b-sqrt(determinant))/(2*a);
	printf("Roots are real and different.");
	printf("\n x1 = %.3f",x1);
	printf("\n x2 = %.3f",x2);
	}
	else if(determinant == 0)
	{
		printf("Roots are real and same.");
		x1=(-b+sqrt(determinant))/(2*a);
		printf("\n x1 = %.3f",x1);
```

```
            printf("\n x2 = %.3f",x2);
        }
        else
        {
            realpart=-b/(2*a);
            imaginarypart=sqrt(-determinant)/(2*a);
            printf("\nRoots are complex and different.");
            printf("\n x1=%.3f+%.3fi",realpart,imaginarypart);
            printf("\n x2=%.3f+%.3fi",realpart,imaginarypart);
        }
        return 0;
}
```

- Write a program to print the following patterns. (a, b ,c at Pg. 59 to 61) (d to h can be done by self)

a. 1
12
123
1234

SOLUTION:

```
#include<stdio.h>
#include<conio.h>
/*print following pattern
1
12
123*/
int main()
{
	int i,j,n;
	printf("Enter the number of rows:");
	scanf("%d",&n);
	//for used as row wise.
	for(i=1;i<=n;i++)
	{
		// for used as column wise.
		for(j=1;j<=i;j++)
		{
			printf("%d",j);
		}
		printf("\n");

	}
	return 0;
}
```

b. 12345
 1234
 123
 12
 1

SOLUTION:

```
#include<stdio.h>
#include<conio.h>
/* print the following pattern
12345
1234
123
12
1*/
int main()
{
	int i,j;
	for(i=5;i>=1;i--)
	{
		for(j=1;j<=i;j++)
		{
			printf("%d",j);
		}
		printf("\n");
	}
	getch();
	return 0;
}
```

c. 1
   21
   321
   4321
   54321

SOLUTION:

```
#include<stdio.h>
#include<conio.h>
/*print the following pattern
1
21
321
4321
54321*/
int main()
{
	int i,j;
	for(i=1;i<=5;i++)
	{
		for(j=i;j>=1;j--)
		{
			printf("%d",j);
		}
		printf("\n");
	}
	return 0;
}
```

d. *

**

***

****

*****

SOLUTION:

```
#include<stdio.h>
#include<conio.h>
int main()
{
   int i, j;
   int n = 5;  // Number of rows

   for(i = 1; i <= n; i++)
        {
      for(j = 1; j <= i; j++)
               {
         printf("*");
      }
      printf("\n");
   }

   return 0;
}
```

e. *****
****
***
**
*

SOLUTION:

```
#include<stdio.h>
#include<conio.h>
/* print the following pattern
*****
****
***
**
* */
int main()
{
        int i, j;
   int n = 5;  // Number of rows

   for(i = n; i >= 1; i--)
        {
      for(j = 1; j <= i; j++)
               {
         printf("*");
      }
      printf("\n");
   }
   return 0;         }
```

f. *
   ***
  *****
 ****

SOLUTION:

```
#include<stdio.h>
#include<conio.h>
int main()
{

   int n = 4;  // Number of rows

   for(int i = 1; i <= n; i++)
{
      // Print spaces
      for(int j = i; j < n; j++)
               {
         printf(" ");
      }
      // Print stars
      for(int k = 1; k <= (2 * i - 1); k++)
               {
         printf("*");
      }
      printf("\n");
   }
   return 0;
}
```

g. 1

2 3

4 5 6

7 8 9 10

11 12 13 14 15

SOLUTION:

```
#include<stdio.h>
#include<conio.h>
/* print the following pattern
1
23
456
78910
1112131415 */
int main()
{
   int n = 5;  // Number of rows
   int num = 1;  // Starting number

   for(int i = 1; i <= n; i++)
        {
      for(int j = 1; j <= i; j++)
                {
         printf("%d ", num);
         num++;
      }
      printf("\n");
   }
   return 0;                }
```

h. $
$$
$$$
$$$$

SOLUTION:

```
#include<stdio.h>
#include<conio.h>
/* print the following pattern
$
$$
$$$
$$$$ */
int main()
{
   int n = 4;  // Number of rows
   for(int i = 1; i <= n; i++)
        {
      for(int j = 1; j <= i; j++)
               {
         printf("$");
      }
      printf("\n");
   }
   return 0;
}
```

- Program for function call by value. (Pg. 67).

SOLUTION:

```
#include<stdio.h>
#include<conio.h>
//program for function call by value
void swap(int num1, int num2)
{
	int temp;
	temp = num1;
	num1 = num2;
	num2 = temp;
}
int main()
{
	int n1=27,n2=11;
	printf("\nBefore swap");
	printf("\nNumber 1:%d",n1);
	printf("\nNumber 2:%d",n2);
	swap(n1,n2);
	printf("\nAfter Swap");
	printf("\nNumber 1:%d",n1);
	printf("\nNumber 2:%d",n2);
	return 0;
}
```

- Program for function for call by reference. (Pg. 67/68).

SOLUTION:

```
#include<stdio.h>
// Program for function for call by reference.
void swap(int *num1,int *num2)
{
	int temp;
	temp = *num1;
	*num1 = *num2;
	*num2 = temp;
}
int main()
{
	int n1=27,n2=11;
	printf("\n before swap");
	printf("\nNumber 1:%d",n1);
	printf("\nNumber 2:%d",n2);
	swap(&n1,&n2);
	printf("\n after swap");
	printf("\nNumber 1:%d",n1);
	printf("\nNumber 2:%d",n2);
	return 0;
}
```

- pr. 6.1) to find the largest value that is stored in the array. pg 72.

SOLUTION:

```
#include<stdio.h>
// to find the largest value that is stored in the array.
int main()
{
	int a[100],max,num,c,pos=1;
	printf("Enter the number of elements in array\n");
	scanf("%d",&num);
	printf("Enter %d integers\n", num);
	for(c=0;c<num;c++)	{
		scanf("%d",&a[c]);	}
	max = a[0];
	for(c=1;c<num;c++)
	{
		if(a[c]>max)
		{
			max=a[c];
			pos=c+1;
		}
	}
	printf("Maximum elements is present at location %d and it's value is %d.\n", pos,max);
	return 0;
}
```

- pr. 6.2) to compute the sum of all elements stored in an array. pg 73.

SOLUTION:

```
 #include<stdio.h>
// to compute the sum of all elements stored in an array.
int main()
{
        int a[5];
        int i,sum=0;
        int *ptr;
        printf("\n Enter 5 Elements:");
        for(i=0;i<5;i++)
                scanf("%d",&a[i]);
                ptr=a;                  //a=&a[0]
                for(i=0;i<5;i++)
                {
                        sum=sum+ *ptr;
                        ptr++;
                }

        printf("The sum of array elements:%d",sum);
        return 0;
}
```

- pr. 6.3) to arrange the 'n' numbers stored in the array in ascending and descending order. pg. 73.

SOLUTION:

```
#include<stdio.h>
//to arrange the 'n' numbers stored in the array in ascending and descending order.
int main()
{
	int a[10],i=0,j=0,n,t;
	printf("\n Enter the number of elements:");
	scanf("%d", &n);
	printf("\n");
	for(i=0;i<n;i++)
	{
		scanf("%d",&a[i]);
	}
	for(j=0;j<(n-1);j++)
	{
		for(i=0;i<(n-1);i++)
		{
			if(a[i] > a[i+1])
			{
				t= a[i];
				a[i]= a[i+1];
				a[i+1]=t;
			}
		}
	}
```

```
        printf("\n Ascending order:");
        for(i=0; i<0; i++)
        {
                printf("%d",a[i]);
        }
        printf("\n Descending order:");
        for(i=n;i>0;i--)
        {
                printf("%d",a[i-1]);
        }
        return 0;
}
```

- pr. 6.4) that performs addition and subtraction of matrices. pg 75.

SOLUTION:

```
#include<stdio.h>
int main()
{
	int i,j,c,r;
	int a[10][10],b[10][10],madd[10][10],msub[20][20];
	printf("\nEnter the value for row and column:");
	scanf("%d %d",&c,&r);
	printf("\n Enter the value for matrix A.\n");
	for(i=0;i<c;i++)
	{
		for(j=0;j<r;j++)
		{
			scanf("\t%d",&a[i][j]);
		}
		printf("\n");
	}
	printf("\n Enter the value for matrix B.\n");
	for(i=0;i<c;i++)
	{
		for(j=0;j<r;j++)
		{
			scanf("\t%d",&b[i][j]);
		}
		printf("\n");
	}
```

```
printf("\n Matrix A:\n");
for(i=0;i<c;i++)
{
	for(j=0;j<r;j++)
	{
		printf("\t%d",a[i][j]);
	}
}
printf("\n");
printf("\n Matrix b:\n");
for(i=0;i<c;i++)
{
	for(j=0;j<r;j++)
	{
		printf("\t%d",b[i][j]);
	}
}
printf("\n");
for(i=0;i<c;i++)
{
	for(j=0;j<r;j++)
	{
		madd[i][j]=a[i][j]+b[i][j];
		msub[i][j]=a[i][j]-b[i][j];
	}
}
printf("\nThe addition matrix is:\n");
for(i=0;i<c;i++)
```

```
        {
                printf("\t%d",madd[i][j]);
        }
        printf("\n");
        printf("\nThe subtraction matrix is:\n");
        for(i=0;i<c;i++)
        {
                printf("\t%d",msub[i][j]);
        }
        printf("\n");
        return 0;
}
```

- pr. 6.5) that performs the multiplication of matrices. pg. 77.

SOLUTION:

```
#include<stdio.h>
int main()
{
	int i,j,c,r,k;
	int a[10][10],b[10][10],mmu[10][10];
	printf("\nEnter the value for row and column:");
	scanf("%d %d",&c,&r);
	printf("\n Enter the value for matrix A.\n");
	for(i=0;i<c;i++)
	{
		for(j=0;j<r;j++)
		{
			scanf("\t%d",&a[i][j]);
		}
		printf("\n");
	}
	printf("\n Enter the value for matrix B.\n");
	for(i=0;i<c;i++)
	{
		for(j=0;j<r;j++)
		{
			scanf("\t%d",&b[i][j]);
		}
```

```
        printf("\n");
}
printf("\n Matrix A:\n");
for(i=0;i<c;i++)
{
        for(j=0;j<r;j++)
        {
                printf("\t%d",a[i][j]);
        }
        printf("\n");
}


printf("\n Matrix b:\n");
for(i=0;i<c;i++)
{
        for(j=0;j<r;j++)
        {
                printf("\t%d",b[i][j]);
        }
        printf("\n");
}

for(i=0;i<c;i++)
{
        for(j=0;j<r;j++)
```

```
            {
                mmu[i][j]=0;
                for(k=0;k<c;k++)
                {
                    mmu[i][j]+=a[i][j]*b[i][j];
                }
            }
        }
        printf("\nThe multiplication matrix is:\n");
        for(i=0;i<c;i++)
        {
            for(j=0;j<r;j++)
            {
                printf("\t%d",mmu[i][j]);
            }
            printf("\n");
        }

        return 0;
}
```

- pr. 7.2) to dereferencing of pointers. pg 83.

SOLUTION:

```
 #include<stdio.h>
int main()
{
        int T, *S;
        T=10;
        S= &T;
        printf("\n%d",*S);              //will give value of T.
        printf("\n%d",*&T);             //will give value of T.
        printf("\n%u",&T);              //will give address of T.
        printf("\n%u",S);               //will give address of T.
        printf("\n%u",&T);              //will give address of S.
        return 0;
}
```

- pr. 7.3) for working of address operator. pg 84.

SOLUTION:

```
#include<stdio.h>
int main()
{
        int T=25;
        printf("\n Value of T is: %d", T);
        printf("\n Value of T is: %u",&T);
        return 0;
}
```

- pr. 7.4) for understanding address operator. pg 84.

SOLUTION:

```
#include<stdio.h>
int main(){
	int S = 5;
	int *myptr;
	myptr = &S;
	printf("\n Address of S :%u",&S);
	printf("\n Value of myptr is :%u",myptr);
	return 0;
}
```

- pg. 7.5) for function pointer. pg 85.

SOLUTION:

```
#include<stdio.h>
int myfunction(int a, int b){
	printf("\n a=%d\n",a);
	printf("\n b=%d\n",b);
	return 0;
}
int main(void){
	int (*myfunctionp)(int,int);
	myfunctionp = myfunction;
	myfunction(2,3);
	myfunctionp(2,3);
	return 0;
}
```