

Part -A

- 1) Write a program to find the largest of three numbers using ternary operators.

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

```
int main()
{
    int a, b, c, largest;

    // Input three numbers
    printf("Enter three numbers: ");
    scanf("%d %d %d", &a, &b, &c);

    // Using nested ternary operator to find the largest number
    largest = (a > b) ? ((a > c) ? a : c) : ((b > c) ? b : c);

    // Output the largest number
    printf("The largest number is: %d\n", largest);

    return 0;
}
```

Out Put:

Enter three numbers: 12 45 23

The largest number is: 45

- 2) To find the roots of a quadratic equation.

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
    int a,b,c,choice;
    double disc,root1,root2,real,img;
    clrscr();
    printf("\n Enter a,b,c : ");
    scanf("%d %d %d",&a,&b,&c);
    disc=(b*b) - (4.0*a*c);
    if(disc > 0)
        choice=1;
    else if(disc < 0)
        choice=2;
    else
        choice=3;
    switch(choice)
    {
```

output:

Enter a,b,c : 1 5 2

Real and Distinct Roots

Root1 = -0.44

Root2 = -4.56

Enter a,b,c : 1 2 3

Roots are complex and imaginary

Root1 = -1.00 +i 1.41

Root2 = -1.00 -i 1.41

```

    case 1:
    {
printf("\n Real and Distinct Roots");
root1=( -b + sqrt(disc) ) / (2.0 * a);
root2=( -b - sqrt(disc) ) / (2.0 * a);
printf("\n Root1= %0.2lf",root1);
printf("\n Root2= %0.2lf",root2);
    }
    break;
case 2:
{
printf("\n Roots are complex and imaginary"); real = -b/(2.0 * a);
img = sqrt(abs(disc))/(2.0 * a);
printf("\n Root1 = %0.2lf +i %0.2lf",real,img); printf("\n Root2 = %0.2lf -i
%0.2lf",real,img);
}
break;

case3:
{
printf("\n Roots are Equal"); root1= -b/(2.0 * a);
root2= -b/(2.0 * a);
printf("\n Root1 = %0.2lf",root1); printf("\n Root2 = %0.2lf",root2);
}
break;
default : printf("\n Invalid inputs");
}
getch();

```

Enter a,b,c : 1 2 1

Roots are equal

Root1 = -1.00

Root2 = -1.00

3) Write a program that uses both recursive and non recursive function.

i) To find factorial of a given number

ii) To solve the tower of hanoi problem.

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

```
// Recursive function to find factorial
```

```
int factorial_recursive(int n)
```

```

{
    if (n == 0 || n == 1)
    {
        return 1;
    }
    return n * factorial_recursive(n - 1);
}

```

```
// Non-recursive (iterative) function to find factorial
```

```
int factorial_non_recursive(int n)
```

```

{
    int result = 1;
    for (int i = 1; i <= n; i++)
    {
        result *= i;
    }
    return result;
}

```

```

}

// Recursive function to solve the Tower of Hanoi problem
void tower_of_hanoi(int n, char source, char destination, char auxiliary)
{
    if (n == 1)
    {
        printf("Move disk 1 from %c to %c\n", source, destination);
        return;
    }
    tower_of_hanoi(n - 1, source, auxiliary, destination);
    printf("Move disk %d from %c to %c\n", n, source, destination);
    tower_of_hanoi(n - 1, auxiliary, destination, source);
}

int main()
{
    int num;

    // Factorial using recursive and non-recursive functions
    printf("Enter a number to find its factorial: ");
    scanf("%d", &num);

    // Recursive factorial
    printf("Factorial of %d using recursion: %d\n", num,
factorial_recursive(num));

    // Non-recursive factorial
    printf("Factorial of %d using iteration: %d\n", num,
factorial_non_recursive(num));

    // Tower of Hanoi problem
    int disks;
    printf("\nEnter the number of disks for the Tower of Hanoi: ");
    scanf("%d", &disks);

    printf("Steps to solve the Tower of Hanoi with %d disks:\n", disks);
    tower_of_hanoi(disks, 'A', 'C', 'B');

    return 0;
}

```

Output:

Enter a number to find its factorial: 5

Factorial of 5 using recursion: 120

Factorial of 5 using iteration: 120

Enter the number of disks for the Tower of Hanoi: 3

Steps to solve the Tower of Hanoi with 3 disks:

Move disk 1 from A to C

Move disk 2 from A to B

Move disk 1 from C to B

Move disk 3 from A to C

Move disk 1 from B to A

Move disk 2 from B to C

Move disk 1 from A to C

4) program to read a number, find the sum of the digits reverse the number and check it for palirome

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int num,orgnum;
    int sum=0,rev=0,rem;
    clrscr();
    printf("\n Enter a number : ");
    scanf("%d",&num);
    orgnum=num;

    while(num>0)
    {
        rem=num%10;
        sum=sum+rem;
        rev=rev*10+rem;
        num=num/10;
    }
    printf("\n Sum of digits = %d",sum);
    printf("\n Reversed number = %d",rev);
    if(orgnum==rev)
        printf("\n Number is a Palindrome");
    else
        printf("\n Number is not a Palindrome");
    getch();
}
```

output:
Enter a number : 1234

Sum of digits = 10
Reversed number= 4321
Number is not a Palindrome

output:
Enter a number : 1221

Sum of digits = 6
Reversed number
=1221

Number is a Palindrome

5) Write a program that uses functions to perform the following:

i) Addition of Two Matrices

ii) Multiplication of Two Matrices

```
#include <stdio.h>

// Function to add two matrices

void addMatrices(int rows, int cols, int matrix1[rows][cols], int matrix2[rows][cols],
int result[rows][cols]) {
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            result[i][j] = matrix1[i][j] + matrix2[i][j];
        }
    }
}
```

// Function to multiply two matrices

```
void multiplyMatrices(int rows1, int cols1, int matrix1[rows1][cols1], int rows2, int cols2, int matrix2[rows2][cols2], int result[rows1][cols2]) {
```

```
    for (int i = 0; i < rows1; i++) {
        for (int j = 0; j < cols2; j++) {
            result[i][j] = 0;
            for (int k = 0; k < cols1; k++) {
                result[i][j] += matrix1[i][k] * matrix2[k][j];
            }
        }
    }
}
```

// Function to print a matrix

```
void printMatrix(int rows, int cols, int matrix[rows][cols]) {
```

```
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            printf("%d ", matrix[i][j]);
        }
        printf("\n");
    }
}
```

// Function to take matrix input from the user

```
void inputMatrix(int rows, int cols, int matrix[rows][cols]) {
```

```
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            printf("Enter element [%d][%d]: ", i + 1, j + 1);
            scanf("%d", &matrix[i][j]);
        }
    }
}
```

```
int main()
```

```

{
    int rows, cols;

    printf("Enter the number of rows and columns for the matrices: ");
    scanf("%d %d", &rows, &cols);

    int matrix1[rows][cols], matrix2[rows][cols], additionResult[rows][cols],
    multiplicationResult[rows][cols];

    printf("\nEnter elements of Matrix 1:\n");
    inputMatrix(rows, cols, matrix1);

    printf("\nEnter elements of Matrix 2:\n");
    inputMatrix(rows, cols, matrix2);

    // Addition of two matrices
    addMatrices(rows, cols, matrix1, matrix2, additionResult);
    printf("\nAddition of Matrix 1 and Matrix 2:\n");
    printMatrix(rows, cols, additionResult);

    // Multiplication of two matrices
    if (rows == cols)
    {
        // Only square matrices are supported for multiplication
        multiplyMatrices(rows, cols, matrix1, rows, cols, matrix2,
        multiplicationResult);
        printf("\nMultiplication of Matrix 1 and Matrix 2:\n");
        printMatrix(rows, cols, multiplicationResult);
    }
    else
    {
        printf("\nMatrix multiplication is not possible for non-square matrices.\n");
    }

    return 0;
}

```

Output:

6) To remove Duplicate Elements in a single dimensional Array

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

```
void main()
```

```
{  
int a[20],n,i,j,ele;
```

```
clrscr();
```

```
printf("\n Enter array limit(n) : ");
```

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

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

```
{
```

```
printf("\n Enter a[%d] : ",i);
```

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

```
}
```

```
//remove duplicate
```

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

```
{
```

```
ele=a[i];
```

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

```
{
```

```
if(ele==a[j] && a[j]!=-111)
```

```
{
```

```
printf("\n %d Duplicate entry!!!",a[j]); a[j]=-111; //set -111 for duplicate entry
```

```
}
```

```
}
```

```
}
```

```
printf("\n Final Array list\n");
```

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

```
{
```

```
if(a[i]!=-111)
```

```
printf("%5d",a[i]);
```

```
}
```

```
getch();
```

```
}
```

output:

Enter array limit(n):5

Enter a[0] : 10

Enter a[1] : 20

Enter a[2] : 10

Enter a[3] : 30

Enter a[4] : 10

10 Duplicate entry!!!

10 Duplicate entry!!!

Final Array list

10 20 30

7) To read a string and to find the number of alphabets, digits, vowels, consonants, spaces and special characters.

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

```
#include<conio.h> #include<ctype.h> void main()
```

```
{
```

```
char str[100],ch;
```

```
int acount=0, dcount=0, vcount=0, ccount=0, scount=0, spcount=0, i=0;
```

```
clrscr();
```

```
printf("\n Enter a string : ");
```

```
gets(str);
```

output:

```

while(str[i]!='\0')
{
    if(isalpha(str[i]))
    {
        acount++;
        ch=tolower(str[i]);
        switch(ch)
        {
            case 'a':
            case 'e':
            case 'i':
            case 'o':
            case 'u': vcount++;
            break; default : ccoun++;;
        }
    }
    else if(isdigit(str[i])) dcount++;
    else if(isspace(str[i])) scount++;
    else
    spcount++;
    i=i++;
}
printf("\n No. of Alphabets = %d",acount); printf("\n No. of Vowels = %d",vcount);
printf("\n No. of Consonants = %d",ccount); printf("\n No. of Spaces = %d",scount);
printf("\n No. of Digits = %d",dcount);
printf("\n No. of Special symbol = %d",spcount);
getch();
}

```

Enter a string : COM_123*012 SIMS

No. of Alphabets = 7
 No. of Vowels = 2
 No. of Consonants = 5
 No. of Spaces = 1
 No. of Digits = 6
 No. of Special symbols = 2

- 8) **c program to input name, marks of 5 subjects of students and display the name of the students, total marks scored ,percentage and class of the result.**

```

#include <stdio.h>
int main()
{
    char name[50];
    int marks[5], total = 0;
    float percentage;
    char result[20];

    // Input student name
    printf("Enter student name: ");
    scanf("%s", name);

    // Input marks for 5 subjects
    printf("Enter marks of 5 subjects:\n");
    for (int i = 0; i < 5; i++) {
        printf("Subject %d: ", i + 1);
        scanf("%d", &marks[i]);
        total += marks[i];
    }

    // Calculate percentage
    percentage = total / 5.0;

```



```

// Determine class result based on percentage
if (percentage >= 75)
{
    sprintf(result, "Distinction");
}
else if (percentage >= 60)
{
    sprintf(result, "First Class");
}
else if (percentage >= 50)
{
    sprintf(result, "Second Class");
}

else if (percentage >= 35)
{
    sprintf(result, "Pass");
}
else
{
    sprintf(result, "Fail");
}

// Display result
printf("\nStudent Name: %s\n", name);
printf("Total Marks: %d\n", total);
printf("Percentage: %.2f%%\n", percentage);
printf("Class Result: %s\n", result);

return 0;
}

```

Output:

Enter student name: John Doe

Enter marks of 5 subjects:

Subject 1: 78

Subject 2: 85

Subject 3: 92

Subject 4: 88

Subject 5: 74

Student Name: John Doe

Total Marks: 417

Percentage: 83.40%

Class Result: Distinction

9) C Program to Construct a Pyramid of Numbers:

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

```

int main() {
    int rows;

    // Input number of rows
    printf("Enter the number of rows: ");
    scanf("%d", &rows);

    // Construct pyramid
    for (int i = 1; i <= rows; i++) {

```

```

        // Print spaces
        for (int j = 1; j <= rows - i; j++) {
            printf(" ");
        }
        // Print numbers
        for (int k = 1; k <= i; k++) {
            printf("%d ", k);
        }
        // Move to the next row
        printf("\n");
    }

    return 0;
}

```

Output:

```

1
1 2
1 2 3
1 2 3 4
1 2 3 4 5

```

10 Program to demonstrate student structure to read & display records of n students

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

```
#include<conio.h>
```

```
struct STUDENT
```

```
{
int regno;
char name[50];
char grade;
```

```
BCA[50];
```

```
void main()
```

```
{
```

```
int n,i;
```

```
clrscr();
```

```
printf("\n Enter student count(n) : ");
```

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

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

```
{
```

```
printf("\n Enter %d student(regno,name,grade) : ",i+1);
```

```
scanf("%d %s %c",&BCA[i].regno, BCA[i].name, &BCA[i].grade);
```

```
}
```

```
printf("\n REGNO \t STUDENT NAME \t GRADE"); for(i=0;i<n;i++)
```

```
{
```

```
printf("\n %d \t%s \t%c", BCA[i].regno, BCA[i].name, BCA[i].grade);
```

```
}
```

```
getch();
```

output:

Enter student count(n) : 3

Enter 1 student(regno,name,grade) : 101 PRANAVI A

Enter 2 student(regno,name,grade) : 102 NAKUL B

Enter 3 student(regno,name,grade) : 103RAMESH C

REGNO	STUDENT NAME	GRADE
101	PRANAVI	A
102	NAKUL	B
103	RAMESH	C

}