

Control Statements

```
/*P5.5 program to find biggest number from three given numbers*/
#include<stdio.h>
int main(void)
{
    int a,b,c,big;
    printf("Enter three numbers : ");
    scanf("%d%d%d",&a,&b,&c);
    if(a>b)
    {
        if(a>c)
            big=a;
        else
            big=c;
    }
    else
    {
        if(b>c)
            big=b;
        else
            big=c;
    }
    printf("Biggest number is %d\n",big);
    return 0;
/*End of main()*/
```

The braces in this if-else structure can be dropped and we can write our code like this:

Now since the value of n is equal to zero, the condition ($n > 0$) becomes false and the loop stops.

```
/*P5.11 Program to find the product of digits of any number*/
#include<stdio.h>
int main(void)
{
    int n,prod=1,rem;
    printf("Enter a number : ");
    scanf("%d",&n);
    while(n>0)
    {
        rem = n%10;      /*taking last digit of n*/
        prod*=rem;
        n/=10;          /*skipping last digit of n*/
    }
    printf("Product of digits = %d\n",prod);
    return 0;
}
```

Output :

Enter a number : 234
Product of digits = 24

The logic of extracting digits is similar to that in earlier programs.

5.6 goto

This is an unconditional control statement that transfers the flow of control to another part of the program. The `goto` statement can be used as-

```
goto label;  
.....  
.....  
label:  
    statement  
.....  
.....
```

Here `label` is any valid C identifier and it is followed by a colon. Whenever the statement `goto label;` is encountered, the control is transferred to the statement that is immediately after the label.

The following program finds whether a number is even or odd. Although it is not a good idea to use `goto` for this purpose, it tells us about the syntax of `goto` statement.

```
/*P5.31 Program to find whether a number is even or odd*/
#include<stdio.h>
int main(void)
{
    int n;
    printf("Enter a number : ");
    scanf("%d",&n);
    if(n%2==0)
        goto even;
    else
        goto odd;
even:
    printf("Number is even\n");
    goto end;
odd:
    printf("Number is odd\n");
    goto end;
end:
    printf("\n");
    return 0;
}
```

Output :

Enter a number : 14

Number is even

```
/*P5.21 Program to generate fibonacci series
1, 1, 2, 3, 5, 8, 13, 34, 55, 89.....
In this series each number is a sum of the previous two numbers*/
#include<stdio.h>
int main(void)
{
    long x,y,z;
    int i,n;
    x=0;
    y=1;
    printf("Enter the number of terms : ");
    scanf("%d",&n);
    printf("%ld ",y);
    for(i=1; i<n; i++)
    {
        z=x+y;
        printf("%ld ",z);
        x=y;
        y=z;
    }
    printf("\n");
    return 0;
}
```

All the three expressions of the for loop are correct. We

body is known as iteration, and the expression is known as loop condition.

```
/*P5.8 Program to print numbers from 1 to 10 using while loop*/
#include<stdio.h>
int main(void)
{
    int i=1;
    while(i<=10)
    {
        printf("%d\t",i);
        i=i+1; /*Statement that changes the value of condition*/
    }
    printf("\n");
    return 0;
}
```

Output:

```
1   2   3   4   5   6   7   8   9   10
```

```
/*P5.9 Program to print numbers in reverse order with a difference of 2*/
#include<stdio.h>
int main(void)
{
    int k=10;
    while(k>=2)
    {
        printf("%d\t",k);
        k=k-2;
    }
    printf("\n");
    return 0;
}
```

Output:

10 8 6 4 2

```
/*P5.10 Program to print the sum of digits of any number*/
```



Ad



v

25 November

Find given element in array

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

```
int main()
{
    int myNumbers[] = {10, 25, 50, 75, 100 ,15,5};
    int input;
    printf("Enter any number: ");
    scanf(" %d", &input);

    for(int i=0; i<7 ; i++)
    {
        if(input == myNumbers[i])
        {
            printf("Your number is available in array");
            return 0;
        }
    }
    printf("Your number is not available in array");
}
```

Find Max and Min element

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

```
int main()
{
    int myNumbers[] = {10, 25, 50, 75, 100, 15, 5};
    int min=myNumbers[0]; //10
    int max=myNumbers[0]; //10

    for(int i=0; i<7; i++)
    {
        if (myNumbers[i] > max)      //10>10 // 25>10 // 50>25 //
        {
            max = myNumbers[i];
        }
        if (myNumbers[i] < min) //10<10 // 25<10 // 50<10 //
        {
            min = myNumbers[i];
        }
    }

    printf(" Max: %d\n", max);
    printf(" Min: %d\n", min);

    return 0;
}
```

8.2 Two Dimensional Arrays

8.2.1 Declaration and Accessing Individual Elements of a 2-D array

The syntax of declaration of a 2-D array is similar to that of 1-D arrays, but here we have two subscripts.

```
data_type array_name [rowsize] [columnsize];
```

Here `rowsize` specifies the number of rows and `columnsize` represents the number of columns in the array. The total number of elements in the array are `rowsize*columnsize`. For example suppose we have an array `arr` declared as-

```
int arr[4][5];
```

Here `arr` is a 2-D array with 4 rows and 5 columns. The individual elements of this array can be accessed by applying two subscripts, where the first subscript denotes the row number and the second subscript denotes the column number. The starting element of this array is `arr[0][0]` and the last element is `arr[3][4]`. The total number of elements in this array is $4*5 = 20$.

	Col 0	Col 1	Col 2	Col 3	Col 4
Row 0	arr[0][0]	arr[0][1]	arr[0][2]	arr[0][3]	arr[0][4]
Row 1	arr[1][0]	arr[1][1]	arr[1][2]	arr[1][3]	arr[1][4]
Row 2	arr[2][0]	arr[2][1]	arr[2][2]	arr[2][3]	arr[2][4]
Row 3	arr[3][0]	arr[3][1]	arr[3][2]	arr[3][3]	arr[3][4]

8.2.2 Processing 2-D Arrays

A 2-D array is also known as a matrix. The next program adds two matrices; the order of both the matrices should be same.

```
/*PS.11 Addition of two matrices*/
#define ROW 3
#define COL 4
#include<stdio.h>
int main(void)
{
    int i,j,mat1[ROW][COL],mat2[ROW][COL],mat3[ROW][COL];
    printf("Enter matrix mat1(%dx%d) row-wise :\n",ROW,COL);
    for(i=0; i<ROW; i++)
        for(j=0; j<COL; j++)
            scanf("%d",&mat1[i][j]);
    printf("Enter matrix mat2(%dx%d) row-wise :\n",ROW,COL);
    for(i=0; i<ROW; i++)
        for(j=0; j<COL; j++)
            scanf("%d",&mat2[i][j] );
    /*Addition*/
    for(i=0; i<ROW; i++)
        for(j=0; j<COL; j++)
            mat3[i][j] = mat1[i][j] + mat2[i][j];
    printf("The resultant matrix mat3 is :\n");
    for(i=0; i<ROW; i++)
    {
        for(j=0; j<COL; j++)
            printf("%5d",mat3[i][j]);
        printf("\n");
    }
    return 0;
}
```

Output :

Enter matrix mat1(3x4) row-wise :

1284

5678

3214

Enter matrix mat2(3x4) row-wise :

2542

1526

9472

The resultant matrix mat3 is.:

3 7 12 6

6 11 9 14

12 6 8 6

Now we will write a program to multiply two matrices. Multiplication of matrices requires that the number of columns of first matrix is equal to the number of rows of second matrix. For example if first matrix is

```

/*P8.12 Multiplication of two matrices*/
#include<stdio.h>
#define ROW1 3
#define COL1 4
#define ROW2 COL1
#define COL2 2
int main(void)
{
    int mat1[ROW1][COL1],mat2[ROW2][COL2],mat3[ROW1][COL2];
    int i,j,k;
    printf("Enter matrix mat1(%dx%d) row-wise :\n",ROW1,COL1);
    for(i=0; i<ROW1; i++)
        for(j=0; j<COL1; j++)
            scanf("%d",&mat1[i][j]);
    printf("Enter matrix mat2(%dx%d) row-wise :\n",ROW2,COL2);
    for(i=0; i<ROW2; i++)
        for(j=0; j<COL2; j++)
            scanf("%d",&mat2[i][j]);
    /*Multiplication*/
    for(i=0; i<ROW1; i++)
        for(j=0; j<COL2; j++)
    {
        mat3[i][j] = 0;
        for(k=0; k<COL1; k++)
            mat3[i][j] += mat1[i][k] * mat2[k][j];
    }
    printf("The Resultant matrix mat3 is :\n");
    for(i=0; i<ROW1; i++)
    {
        for(j=0; j<COL2; j++)
            printf("%5d",mat3[i][j]);
        printf("\n");
    }
    return 0;
}

```

Output :

Enter matrix mat1(3x4)row-wise :

2 1 4 3

5 2 7 1

3 1 4 2

Enter matrix mat2(4x2)row-wise :

1 2

3 4

2 5

6 2

The Resultant matrix mat3 is :

31 34

31 55

26 34

The next program finds out the transpose of a matrix T

Problem 1

Write a program to enter a string and print it in reverse order.

```
/*P10.29 Program to enter a string and print it in reverse order*/
#include<stdio.h>
#include<string.h>
int main(void)
{
    char str[50];
    int len;
    printf("Enter a string : ");
    gets(str);
    for(len=strlen(str)-1; len>=0; len--)
        printf("%c",str[len]);
    return 0;
}
```