**Programming**

**what is programming.?**

* **Computer only understands 0 & 1. (on & off)**
* **In word you drag select bold style color texts, it automatically happens.**
* **Computer understands the above process which only know 0 & 1.**
* **Programmers have written codes – some instructions which make the above process happen.**
* **When we give a set of instructions to computer, it will work according to it.**
* **Computer hardware has no brain to think, but we make it happen using coding.**
* **Eg2: WhatsApp chat**
* **Programming- set of instructions fed to computer, in return computer perform some actions or tasks.**
* **Computer must understand these set of instructions, so these codes must be in a language that a computer understands.**
* **Programming language—language that computer knows.**
* **We code not for us.**
* **Developers will teach the system what to do, how to do for users.**
* **We have some users, to provide solution for our users we have a computer system.**
* **We(developer) will always stand with the system.**
* **Computer cannot take decision on its own.**
* **Computer only know 1 and 0.**
* **Eg calculator. We get inputs, system process the inputs according to the instructions given and provide output.**
* **What computer thinks is our thinking.**
* **Software related issues – not computer, it is always us who write the code**

**COMPILER:**

* **we write code human readable language**
* **to convert these codes to machine understanding language ie, 0 &1 after checking the written code.**
* **Converts to machine language..**
* **We run the program using these machine language.**
* **User gets these machine language code to run.**

**SOFTWARE DEVELOPMENT KIT SDK**

* **We can do coding only by installing sdk**
* **Each language has its own sdk depending on os.**
* **Contains libraries, compiler kits, etc,.**

**INTEGRATED DEVELOPMENT ENVIRONMENT IDE**

* **We can code using nodepad.**
* **Prompts code, provides suggestion, corrections, run or compile by just a click.**
* **There wont be essential tools to make coding simple in notepad, wont correct spelling mistakes.**
* **Works as a text editor.**

**Console output: where we get output while coding.**

**Coding screen: where we write code.**

**User interacts with the system using console output; develops interacts using coding screen.**

**Initially developer interacts with both.**

**We always code for users. So, we should deliver what the user wants.**

**C-programming**

**First compile the code and convert to binary**

**First you need to build the project**

**Just run the binary file.**

1. **Printf**

to display/deliver something to user – output statement

syntax: printf(“Hello”);

output: Hello gets printed/displayed on console.

We can write paragraph too inside “ ”

Program1:

code

#include<stdio.h>

*int* main (*void*){

    printf("Hi Welcome");

}

Output

Hi Welcome

1. INPUT STATEMENTS

* To get values or anything from user through console.
* When we get something from the user we need to store it
* ***VARIABLE :***  where we store data obtained from user, we can reuse it.

Creating a variable.

* We should name a variable.
* Name can be anything {has some rules}
* We can give any name to the variable of our choice.
* We should initialize the variable.
* We should check the type of the variable
* Integer = 4,3,56,7777,0
* Float=12.34,23.44,2124.3
* **Character=a,b,c,f,e,r**
* Basic data type-int,float,char

DECLARATION

* When we use a variable, we should declare that variable.
* Eg; int a;

INITIALIZATION

* We may not get data from the user every time.
* We assume a value and store it in a variable.
* Syntax: int a=5;
* Assigning a value to a variable.

Semi-colon

* We use semi-colon at the end of each line code in c.

TO GET VALUE FROM USER

* Syntax: scanf(“%d”,&a);
* If we want to get an int value from the user – we use %d.
* If we want to get a float value from the user – we use %f.
* If we want to get a character value from the user – we use %c.
* The value we get from the user must be stored in a variable, so we need to specify the variable name after “” with an &.
* In above example ,a is the variable.

We should display what we need from the user.

Output a statement with a value.

Printf(“entered number is %d”,a);

* Here %d is an escape character, it does not get displayed as it is.
* %d tells the system that it has to display an integer number.
* a is a variable where the value is stored and to be displayed.

Program2

Code

#include<stdio.h>

*int* main (*void*){

*int* a;

    printf("Enter a number");

    scanf("%d",&a);

    printf("Entered value is: %d",a);

}

Output:

Enter a number13

Entered value is: 13

## instead of putting %d inside double quote, if we put a we get output as

Enter a number13

Entered value is: a

Program3

Sum of two numbers (input from user)

* display in console to enter two numbers.
* get two values from user
* store these two values in variables.
* Find sum of these two variables and store the sum in another variable.
* Display the sum

code

#include<stdio.h>

*int* main(*void*)

{

*int* a,b,sum;

printf("Enter two numbers: ");

scanf("%d%d",&a,&b);

sum=a+b;

printf("sum of %d and %d is %d",a,b,sum);

}

Output:

Enter two numbers: 13

13

sum of 13 and 13 is 26

#%d inside “ ” takes the respective values outside.

OPERATORS

+,=,-,\*,/, etc,.

1. Arithmetic operators. + - / \* %
2. Relational operators ==,<,>,<=,>=
3. Logical operator &&, ||, !
4. Assignment operator =, +=, -=

// = means **ASSIGN**

**// a==b** means **a** equal to **b**

**PROGRAM4**

**Average of three numbers:**

**Code**

#include<stdio.h>

*int* main(*void*){

*int* num1,num2,num3;

*float* avg;

    printf("Enter 3 numbers: ");

    scanf("%d%d%d",&num1,&num2,&num3);

    avg=(num1+num2+num3)/3;

    printf("average of %d %d %d is %f",num1,num2,num3,avg);

}

Output

Enter 3 numbers: 45

61

47

average of 45 61 47 is 51.000000

PROGRAM 5

Swapping values of two variables

#include<stdio.h>

*int* main(*void*){

*int* a=10,b=20,temp;

    temp=a;

    a=b;

    b=temp;

    printf("a: %d \n b:%d",a,b);

}

Output

a: 20

b:10

method2 for swapping

#include<stdio.h>

*int* main(*void*){

*int* a,b;

    a=10;

    b=13;

    a=a+b;

    b=a-b;

    a=a-b;

    printf("a: %d \n b:%d",a,b);

}

Output

a: 13

b:10

program6 Accept a char input from the user and display it on the console

Code

#include<stdio.h>

*int* main(*void*){

*char* a[50];

    printf("Enter a word");

    scanf("%s", a);

    printf("Your name is %s",a);

}

Output

Enter a word MESSI

Your name is MESSI

Program 7: Accept two inputs from the user and output its sum

|  |  |
| --- | --- |
| **Variable** | **Data Type** |
| Number 1 | Integer |
| Number 2 | Float |
| Sum | Float |

CODE:

#include<stdio.h>

*int* main(*void*){

*float* a,sum;

*int* b;

    printf("Enter a number: ");

    scanf("%d",&b);

    printf("Enter a decimal: ");

    scanf("%f",&a);

    sum=a+b;

    printf("Sum of %d and %f is %f",b,a,sum);

}

OUTPUT:

Enter a number: 13

Enter a decimal: 13.13

Sum of 13 and 13.130000 is 26.130001

PROGRAM 8: Write a program to find the simple interest.

* 1. Program should accept 3 inputs from the user and calculate simple interest for the given inputs. Formula: SI=(P\*R\*n)/100)

|  |  |
| --- | --- |
| **Variable** | **Data Type** |
| Principal amount (P) | Integer |
| Interest rate (R) | Float |
| Number of years (n) | Float |
| Simple Interest (SI) | Float |

Code:

#include<stdio.h>

*int* main(*void*){

*int* principle;

*float* rate,interest,time;

    printf("Enter the principal Amt: ");

    scanf("%d",&principle);

    printf("Enter the rate of Interest: ");

    scanf("%f",&rate);

    printf("Enter the time Period: ");

    scanf("%f",&time);

    interest=(principle\*rate\*time),100;

    printf("Your simple interest is: %f",interest);

}

Output :

Enter the principal Amt: 100

Enter the rate of Interest: 1.5

Enter the time Period: 1.9

Your simple interest is: 285.000000

CONTROL STATEMENTS

* Computer has no ability to take self decisions.
* It only know 0 and 1.
* Help computer to make decisions.
* We will instruct computer to take instructions.

1. If else

If

(condition)

T

**F**

* There will be no condition in else.
* Checks whether the condition in if is true or else gets executed.

**Program9 :** find whether the entered number is negative or positive

Code:

#include<stdio.h>

*int* main(*void*){

*int* a;

    printf("Enter a number: ");

    scanf("%d",&a);

    if (a<0)

    {

        /\* code \*/

        printf("%d is negative number",a);

    }

    else

    {

        printf("%d is positive",a);

    }

}

Output:

Enter a number: 13

13 is positive

Enter a number: -112

-112 is negative number

Program10 : check largest among two numbers

Code:

#include<stdio.h>

*int* main(*void*){

*int* a,b;

    printf("Enter two numbers: ");

    scanf("%d%d",&a,&b);

    if (a>b)

    {

        printf("%d is largest",a);

    }else

    {

        printf("%d is largest",b);

    }

}

Output:

Enter two numbers: 155

45

155 is largest

Enter two numbers: 12

15

15 is largest

**NESTED IF**

* if condition inside an if.
* Eg: check greatest among 3 numbers.

**Program11:**  Check largest among 3 numbers using nested if

Code

#include<stdio.h>

*int* main(*void*){

*int* a,b,c;

    printf("Enter 3 numbers:");

    scanf("%d%d%d",&a,&b,&c);

    if (a>b)

    {

        if (a>c)

        {

            printf("%d is largest",a);

        }

        else{

            printf("%d is largest",c);

        }

    }

    else

    {

        if (b>c)

    {

        printf("%d is largest",b);

    }

    else{

        printf("%d is largest",c);

    }

    }

}

Output:

Enter 3 numbers:11

9

5

11 is largest

Enter 3 numbers:1

3

9

9 is largest

Enter 3 numbers:12

13

11

13 is largest

Else if ladder

* When there exist more than two conditions to check.
* Many else if condition
* \n == new line ; escape sequence;
* == means equal to
* Checks each and every condition.

Program12:

Code:

#include<stdio.h>

*int* main(*void*){

*int* a,b,ch;

    printf("Enter two numbers");

    scanf("%d%d",&a,&b);

    printf("Enter 1 for Addition. \nEnter 2 for Subtraction \nEnter 3 for Multiplication \nEnter 4 Division");

    printf("Enter your choice");

    scanf("%d",&ch);

    if (ch==1)

    {

*int* sum=a+b;

        printf("Sum of %d and %d is %d",a,b,sum);

    }

    else if (ch==2)

    {

*int* diff=a-b;

        printf("Diff is %d and %d is %d",a,b,diff);

    }

    else if (ch==3)

    {

*int* pro=a\*b;

        printf("Product is %d and %d is %d",a,b,pro);

    }

    else if (ch==4)

    {

*float* quo=a/b;

        printf("Quotient is %d and %d is %f",a,b,quo);

    }

    else

    {

        printf("Wrong Entry..::");

    }

}

Output:

Enter two numbers13

23

Enter 1 for Addition.

Enter 2 for Subtraction

Enter 3 for Multiplication

Enter 4 DivisionEnter your choice2

Diff is 13 and 23 is -10

Enter two numbers12

123

Enter 1 for Addition.

Enter 2 for Subtraction

Enter 3 for Multiplication

Enter 4 DivisionEnter your choice1

Sum of 12 and 123 is 135

Enter two numbers13

13

Enter 1 for Addition.

Enter 2 for Subtraction

Enter 3 for Multiplication

Enter 4 DivisionEnter your choice5

Wrong Entry..::

Enter two numbers13

3

Enter 1 for Addition.

Enter 2 for Subtraction

Enter 3 for Multiplication

Enter 4 Division

Enter your choice4

Quotient is 13 and 3 is 4.000000

**Switch Case**

* Greater performance
* Put break; at the end of each case.
* Break is to exist from that respective case
* Unlimited number of cases.
* There will be a default case inside switch to execute else case.
* Goes directly to that case

Program13:

Code

#include<stdio.h>

*int* main(*void*){

*int* cho;

    printf("1. Porotta \n2. Biriyani \n3. Mandhi \n4.Rice");

    printf("\n Enter your choice");

    scanf("%d",&cho);

    switch (cho)

    {

    case 1:

        printf("you have entered porotta");

        break;

    case 2:

        printf("You have enetered Biriyani");

        break;

    case 3:

        printf("You have entered Mandhi");

        break;

    case 4:

        printf("You have entered Rice");

        break;

    default:

        printf("Wrong Entry");

        break;

    }

}

Output:

1. Porotta

2. Biriyani

3. Mandhi

4.Rice

Enter your choice43

Wrong Entry

1. Porotta

2. Biriyani

3. Mandhi

4.Rice

Enter your choice1

you have entered porotta

Program14: Write a program to check whether a student has passed or failed in a subject after he or she enters their mark (pass mark for a subject is 50 out of 100).

Program should accept an input from the user and output a message as “Passed” or “Failed”

|  |  |
| --- | --- |
| **Variable** | **Data type** |
| mark | float |

code:

#include<stdio.h>

*int* main(*void*){

*float* mark;

    printf("Enter your marks in 100: ");

    scanf("%f",&mark);

    if (mark>50)

    {

        printf("You have passed");

    }

    else

    {

        printf("You have failed");

    }

}

Output:

Enter your marks in 100: 12

You are failed

Enter your marks in 100: 51

You have passed

Program15: Write a program to show the grade obtained by a student after he/she enters their total mark percentage.

Program should accept an input from the user and display their grade as follows

|  |  |
| --- | --- |
| **Mark** | **Grade** |
| < 90 | A |
| 80-89 | B |
| 70-79 | C |
| 60-69 | D |
| 50-59 | E |
| < 50 | Failed |

|  |  |
| --- | --- |
| **Variable** | **Data type** |
| Total mark | float |

code:

#include<stdio.h>

*int* main(*void*){

*float* per;

    printf("Enter Your toatal mark percentage:  ");

    scanf("%f",&per);

    if (per>90)

    {

        printf("Grade is A");

    }

    else if (per>80 & per<90 )

    {

        printf("Grade is B");

    }

    else if (per>70 & per<80)

    {

        printf("Grade is c");

    }

    else if (per>60 & per<70)

    {

        printf("Grade is D");

    }

    else if (per>50 & per<60)

    {

        printf("Grade is E");

    }

    else{

        printf("Failed..!");

    }

}

Output:

Enter Your toatal mark percentage: 56

Grade is E

Enter Your toatal mark percentage: 99

Grade is A

Program17: Using the ‘switch case’ write a program to accept an input number from the user and output the day as follows.

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 | Sunday |
| 2 | Monday |
| 3 | Tuesday |
| 4 | Wednesday |
| 5 | Thursday |
| 6 | Friday |
| 7 | Saturday |
| Any other input | Invalid Entry |

Code:

#include<stdio.h>

*int* main(*void*){

*int* num;

    printf("Enter a number 1 - 7: ");

    scanf("%d",&num);

    switch (num)

    {

    case 1:

        printf("Sunday");

        break;

    case 2:

        printf("Monday");

        break;

    case 3:

        printf("Tuesday");

        break;

    case 4:

        printf("Wednesday");

        break;

    case 5:

        printf("Thrusday");

        break;

    case 6:

        printf("Friday");

        break;

    case 7:

        printf("Saturday");

        break;

    default:

    printf("Invalid Entry");

        break;

    }

}

Output:

Enter Your toatal mark percentage: 66

Grade is D

Enter a number 1 - 7: 66

Invalid Entry

Enter a number 1 - 7: 5

Thrusday

**Pre-increment**

* ++a 🡺 a=a+1
* If a=10; b=++a 🡺 b=11, a=11
* First the value gets incremented.

**Post-increment**

* a++ 🡺 a=a+1
* if a=10; b=a++ 🡺 b=10,a=11
* the value gets incremented after .

##Similarly there is post-decrement and pre-decrement

**LOOP**

* if you need to print numbers from 1 to 5 you will use printf 5 times.
* Same instruction to execute many times.

1. For loop
2. While loop
3. Do while

FOR LOOP

For(initialization;condition;updation){

--------------------------------

--------------------------------

}

* Initialize a value =🡺initialization
* Set a condition
* Then increment/decrement the value
* When the condition is true, it executes what is inside the for loop
* Then the value get updated.
* Then it goes back to check whether the updated value satisfies the condition.
* Then it executes the statements inside for loop.
* This process continues till the condition is fulfilled.
* When the condition fail to satisfy, it jumps out of the loop.

Program18: to print numbers from 0 to 10

Code:

#include<stdio.h>

*int* main(*void*){

*int* i;

    for ( i = 0; i <= 10; i++)

    {

        printf("%d\n",i);

    }

}

Output:

0

1

2

3

4

5

6

7

8

9

10

Program19: sum of n numbers

Code:

#include<stdio.h>

*int* main(*void*){

*int* i,sum,n;

    printf("Enter a number");

    scanf("%d",&n);

    sum=0;

    for ( i = 1; i <= n; i++)

    {

        sum=sum+i;

    }

    printf("Sum is %d",sum);

}

Output:

Enter a number3

Sum is 3

Enter a number5

Sum is 15

Program20: print even numbers within a limit.

Code:

#include<stdio.h>

*int* main(*void*){

*int* i,n;

    printf("Enter a limit");

    scanf("%d",&n);

    for ( i = 2; i <=n; i++)

    {

        if (i%2==0)

        {

            printf("%d\n",i);

        }

    }

}

Output:

Enter a limit20

2

4

6

8

10

12

14

16

18

20

Program21: Find prime number

Code:

#include<stdio.h>

*int* main(*void*){

*int* num,i;

*int* flag=0;

    printf("Enter a number");

    scanf("%d",&num);

    for ( i = 2; i <num; i++)

    {

        if (num%i==0)

        {

           flag=1;

           break;

        }

    }

    if (flag==1)

    {

        printf("%d is not prime",num);

    }

    else

    {

        printf("%d is prime",num);

    }

}

Output

Enter a number

45

45 is not prime

Enter a number8

8 is not prime

Nested Loop

= Loop inside a loop.

Program22: print right triangle

Code:

#include<stdio.h>

*int* main(*void*){

*int* i,j,a;

    printf("Enter a limit ");

    scanf("%d",&a);

    for ( i = 1; i <=a; i++)

    {

        for ( j = 0; j < i; j++)

        {

            printf("\*");

        }

        printf("\n");

    }

}

Output:

Enter a limit 5

\*

\*\*

\*\*\*

\*\*\*\*

BREAK & CONTINUE

* break;

it jumps out of the loop.

* Continue;

Won’t execute the statements after continue but will execute the loop.

Break;

#include<stdio.h>

*int* main(*void*){

*int* i;

    for ( i = 0; i < 10; i++)

    {

        printf("hi ");

        if (i==5)

        {

            break;

        }

        printf("Hello\n");

    }

    printf("Finished");

}

Output:

hi Hello

hi Hello

hi Hello

hi Hello

hi Hello

hi Finished

continue;

#include<stdio.h>

*int* main(*void*){

*int* i;

    for ( i = 0; i < 10; i++)

    {

        printf("hi ");

        if (i==5)

        {

            continue;

        }

        printf("Hello\n");

    }

    printf("Finished");

}

Output:

hi Hello

hi Hello

hi Hello

hi Hello

hi Hello

hi

hi Hello

hi Hello

hi Hello

hi Hello

Finished

Program23: Write a program to print the multiplication table of given number

Accept an input from the user and display its multiplication table

Eg:

**Output**: Enter a number

**Input**: 5

**Output**:

1 x 5 = 5

2 x 5 = 10

3 x 5 = 15

4 x 5 = 20

5 x 5 = 25

6 x 5 = 30

7 x 5 = 35

8 x 5 = 40

9 x 5 = 45

10 x 5 = 50

Code:

#include<stdio.h>

*int* main(*void*){

*int* num,mul,i;

    printf("Enter a number ");

    scanf("%d",&num);

    for ( i = 1; i <=10; i++)

    {

        mul=num\*i;

        printf("%d\*%d=%d\n",num,i,mul);

    }

}

Output:

Enter a number 3

3\*1=3

3\*2=6

3\*3=9

3\*4=12

3\*5=15

3\*6=18

3\*7=21

3\*8=24

3\*9=27

3\*10=30

Program24: Write a program to find the sum of all the odd numbers for a given limit

Program should accept an input as limit from the user and display the sum of all the odd numbers within that limit

For example if the input limit is 10 then the result is 1+3+5+7+9 = 25

**Output**: Enter a limit

**Input**: 10

**Output**: Sum of odd numbers = 25

**Code**:

#include<stdio.h>

*int* main(*void*){

*int* num,i,sum;

    printf("Enter a limit:");

    scanf("%d",&num);

    sum=0;

    for ( i = 1; i <=num ; i+=2)

    {

        sum=sum+i;

    }

    printf("sum is %d",sum);

}

Output:

Enter a limit:100

sum is 2500

Enter a limit:10

sum is 25

program25: Write a program to print the following pattern (**hint**: use nested loop)

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

Code:

#include<stdio.h>

*int* main(*void*){

*int* n,i,j;

    printf(“Enter a limit”);

    scanf(“%d”,&n);

    for ( i = 1; i <=n; i++)

    {

        for ( j = 1; j <= i; j++)

        {

            printf(“%d “,j);

        }

        printf(“\n”);

    }

}

Output:

Enter a limit5

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

ARRAYS

* When we are dealing with 2 or 3 values, we store it inside variables , datatypes being int , float, char.
* When we have thousands of data to store it is not possible to declare thousands of variables.
* It would be a task to do any operation with that variables.
* So we will store a group of data having same datatype onto a variable.
* Array=group of similar type of data.
* Position start with index 1.
* Data in each position
* Syntax int a[5]; array of length 5.
* A=10; assigning a value to a;{variable}
* But in array we assign a value to its index position.
* Ie, a[1]=23; means 23 get stored in 1st index (2nd position) in that array.
* When we get data from the user and to store it in an array we need to specify the position (index).
* We use scanf to get data from the user.
* We need to specify the position to which the data is to be stored.
* Scanf(“%d”,&a[4]); 🡺 the user entered data get stored in 4th index position of array a;
* When we have to store multiple data to an array, we need to type scanf that number of times.
* But in that case we use loops.
* Cycle that repeat many times🡺 Loops ; cycle that repeats for many times to execute the same task.
* We use for loop to loop through all the indices.
* The same applicable while printing every/each data in an array.

Program 26: Array input & output

Code:

#include<stdio.h>

*int* main(){

*int* i,a[6];

    printf("Enter 5 numbers.");

    for ( i = 0; i < 5; i++)

    {

        scanf("%d",&a[i]);

    }

    printf("your array is:\t");

    for ( i = 0; i < 5; i++)

    {

        printf("%d\t",a[i]);

    }

}

Output:

Enter 5 numbers.12

10

22

34

55

your array is: 12 10 22 34 55

## \t = escape sequence inserts a tab space

Program 28: array input & output where user specifies the length.

#include<stdio.h>

*int* main(){

*int* i,d;

    printf("Enter the length of array: ");

    scanf("%d",&d);

    printf("Enter values to array. ");

*int* a[d];

    for ( i = 0; i < d; i++)

    {

        scanf("%d",&a[i]);

    }

    printf("Entered values are:\t");

    for ( i = 0; i < d; i++)

    {

        printf("%d\t",a[i]);

    }

}

Output:

Enter the length of array: 7

Enter values to array. 1

23

21

122

1

45

768

Entered values are: 1 23 21 122 1 45 768

## while initializing array put a max. value.

## ask user for array size.

## use for loop to get data and to print it out.

Program 29: sum of elements in an array.

#include<stdio.h>

*int* main(){

*int* l,i,a[100];

    printf("Enter array limit: ");

    scanf("%d",&l);

    printf("Enter array values:");

    for ( i = 0; i < l; i++)

    {

        scanf("%d",&a[i]);

    }

*int* sum=0;

    printf("Entered values are:\t");

    for ( i = 0; i < l; i++)

    {

        printf("%d\t",a[i]);

    }

    for ( i = 0; i < l; i++)

    {

        sum=sum+a[i];

    }

    printf("\n");

    printf("sum of array is %d",sum);

}

Output:

Enter array limit: 6

Enter array values:12

34

54

55

23

1

Entered values are: 12 34 54 55 23 1

sum of array is 179

program 31: search an element in an array.

#include<stdio.h>

*int* main(){

*int* i,l,sch,a[100],flag=0;

    printf("Enter array limi");

    scanf("%d",&l);

    printf("Enter values:");

    for ( i = 0; i < l; i++)

    {

        scanf("%d",&a[i]);

    }

    printf("Enter a value to search");

    scanf("%d",&sch);

    for ( i = 0; i < l; i++)

    {

        if (sch==a[i])

        {

            flag=1;

            break;

        }

    }

    if (flag==1)

    {

       printf("Value found at %d position",i+1);

    }

    else

    {

        printf("Value not found.");

    }

}

Output:

Enter array limi4

Enter values:12

33

43

23

Enter a value to search23

Value found at 4 position

Enter array limi4

Enter values:12

33

34

23

Enter a value to search66

Value not found.

SORTING AN ARRAY (selection sort)

* First index is locked (not value)
* Then it compares with the second
* If it to be changed,we will swap them.
* There will be two loops.
* One to select position,

Program 32 ascending order selection sort

#include<stdio.h>

*int* main(){

*int* i,j,temp,l,a[100];

    printf("Enter array limit: ");

    scanf("%d",&l);

    printf("Enter values");

    for ( i = 0; i < l; i++)

    {

        scanf("%d",&a[i]);

    }

    for ( i = 0; i < l-1; i++)

    {

        for ( j = i+1; j < l ; j++)

        {

           if (a[i]>a[j])

           {

               temp=a[i];

               a[i]=a[j];

               a[j]=temp;

           }

        }

    }

    printf("Array after sorting  ");

    for ( i = 0; i < l; i++)

    {

        printf("%d\t",a[i]);

    }

}

Output:

Enter array limit: 4

Enter values12

45

32

99

Array after sorting 12 32 45 99

Program 33: descending order.

#include<stdio.h>

*int* main(){

*int* i,j,a[100],l,temp;

    printf("Enter array limit");

    scanf("%d",&l);

    printf("Enter your values: ");

    for ( i = 0; i < l; i++)

    {

        scanf("%d",&a[i]);

    }

    for ( i = 0; i < l-1; i++)

    {

        for ( j = i+1; j < l; j++)

        {

            if(a[i]<a[j]){

                temp=a[i];

                a[i]=a[j];

                a[j]=temp;

            }

        }

    }

    printf("array in descending order\t");

    for ( i = 0; i < l; i++)

    {

        printf("%d\t",a[i]);

    }

}

Output:

Enter array limit5

Enter your values: 12

13

79

54

4

array in descending order 79 54 13 12 4