

# PROGRAMMING IN C

**Drishya Neupane**

BE Software

Roll no: 201648

**Ms.Rabina Chaudhary**

Programming in C

Software Department

## **Practical No: 2**

**Title:** TO LEARN THE CONDITIONAL STATEMENTS

**Objective:**

1. Understanding the conditional statements.
2. Understanding proper use and utilization of if, if-else, nested if-else.
3. Using conditional statements to solve conditional problems

**Theory:**

- Conditional Statements/Decision Control Instructions: Conditional Statements are used to make decisions based on the conditions. They execute sequentially when there is no condition around the statements. i.e. if, if-else, nested if-else, nested else-if
- if statement: If statement is the most basic conditional statement. It is always used with a condition and evaluates statements to either true or false. It is used as
  - If (condition is true)  
execute this statement;
- if-else statement: The if statement does nothing when the condition is false. In such case, the if-else statement is used. It is used as
  - If (condition is true)  
execute this statement;  
else  
execute this statement;
- Nested if-else: If multiple conditions are required then nested if-else is used. So using if-else construct within if/or statement is called nesting. It is used as
  - If (condition is true)  
execute this statement;  
else

[ if(condition is true)  
execute this statement  
else

Example:

```
#include<stdio.h>

int main() {

int a;

printf("Enter a number");

scanf("%d",&a);

if (a==0)

    printf("You just typed the number 0");

else

{ if (a > 0)

    printf("You just typed a number greater than 0");

else

    printf("You just typed a number less than 0");

}

printf("

return 0;

}
```

(Here nested if else is used to state w condition whether the number is 0, less than 0 or greater than it)

OUTPUT

Enter a number: -1

You just typed a number less than 0

### Practical:

**1) What will be the value of A and B on each line? A=10, B= 5; A=++A + ++A; B=A-- - --B; Also explain the operations.**

### Algorithm:

Step 1: Start

Step 2: Declare variable A, B

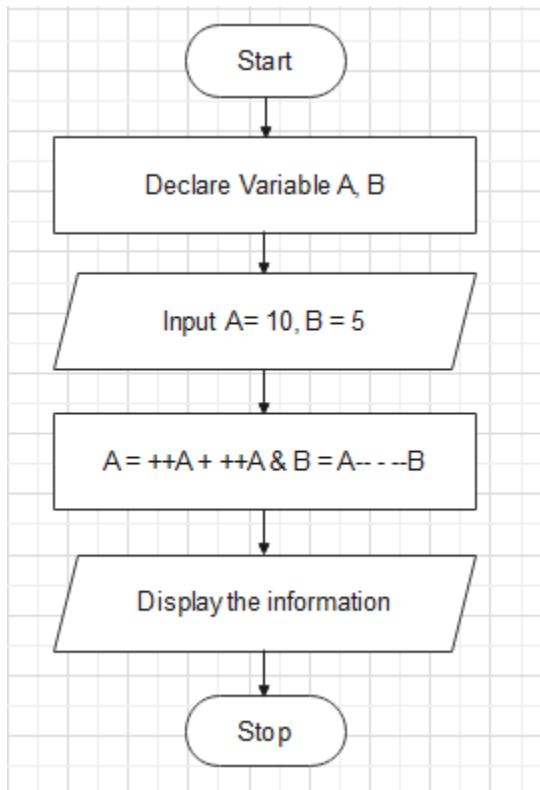
Step 3: Input A = 10, B = 5

Step 4: Perform A = ++A + ++A & B = A-- - --B

Step 5: Display the information

Step 6: Stop

### Flowchart:



### Program:

```
//Drishya Neupane 201648

//Question No 1

#include<stdio.h>
int main(){
    int A=10, B= 5;
    A=++A + ++A;
    B=A-- - --B;
    printf("A = %d , B = %d",A,B);
return 0;
}
```

### Output:

A screenshot of a Windows command prompt window. The title bar at the top reads "C:\Users\user\Desktop\C programming\Project 2\C\2.1.exe". The main area of the window displays the output of a C program: "A = 23 , B = 20", followed by "Process returned 0 (0x0) execution time : 0.065 s", and finally "Press any key to continue." The text is white on a black background.

Here, the final value of A is displayed as 23 and B is displayed as 20

We know A++ will increment the value of A and return it to A

Where ++A will increment the value of A and replace its new value

i.e ++A = 11 (New value of A= 11) & A++ = 11+1 = 12

Then, A = 12 + 11 = 23

Similarly for B,

A--

**2) Write a program to find out whether the entered number is odd or even using conditional operator.**

**Algorithm:**

Step 1: Start

Step 2: Declare variable X

Step 3: Display "Input a number"

Step 4: Enter the number

Step 5: Check if  $X \% 2 = 0$

Step 5.1: If Step 5 is true

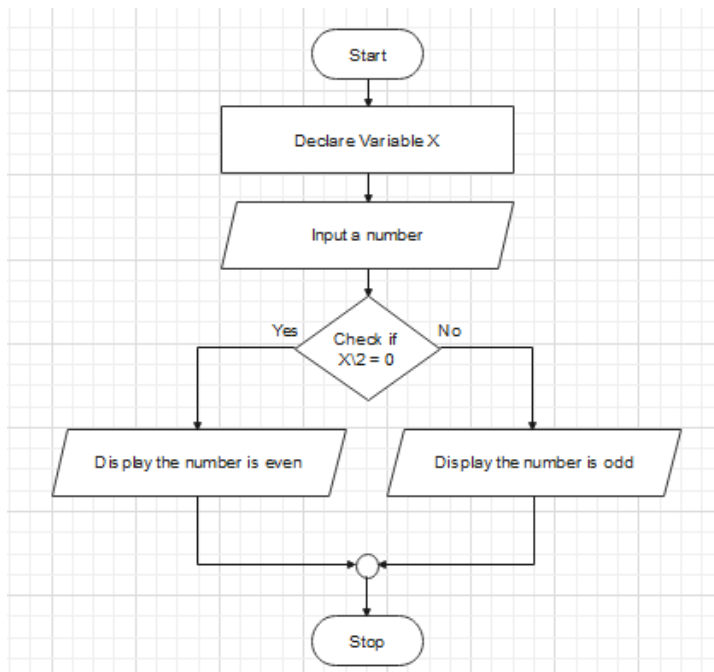
Step 5.2: Display the number is even

Step 5.1: If Step 5 is false

Step 5.2: Display the number is odd

Step 6: Stop

**Flowchart:**



## Program:

```
//Drishya Neupane 201648  
  
//Question No 2  
  
#include<stdio.h>  
int main() {  
    int X;  
    printf("Input a number: ");  
    scanf("%d", &X);  
  
    if (X%2 == 0)  
        //Here a modulus is taken with remainder 0 to find number exactly divisible by 2 i.e even  
        printf("The number is even");  
    else  
        printf("The number is odd");  
    return 0;  
}
```

## Output:

```
Input a number: 987  
The number is odd  
Process returned 0 (0x0)   execution time : 2.134 s  
Press any key to continue.
```

```
Input a number: 986  
The number is even  
Process returned 0 (0x0)   execution time : 1.861 s  
Press any key to continue.
```

**3) Write a program to compare three numbers and to find the maximum out of those numbers.**

**Algorithm:**

Step 1: Start

Step 2: Declare variable a,b,c

Step 3: Display "Input the value of a, b, c"

Step 4: Input the values of a, b, c

Step 5: Check  $a > b$

    Step 5.1: If Step 5 is true

        Step 5.1.1: Check if  $a > c$

            Step 5.1.1.1: If Step 5.1.1 is true

                Step 5.1.1.1.1: Display a is greatest

            Step 5.1.1.2: If Step 5.1.1 is false

                Step 5.1.1.2.1: Display c is greatest

    Step 5.2: If Step 5 is false

        Step 5.2.1: Check if  $b > c$

            Step 5.1.1.1: If Step 5.1.1 is true

                Step 5.1.1.1.1: Display b is greatest

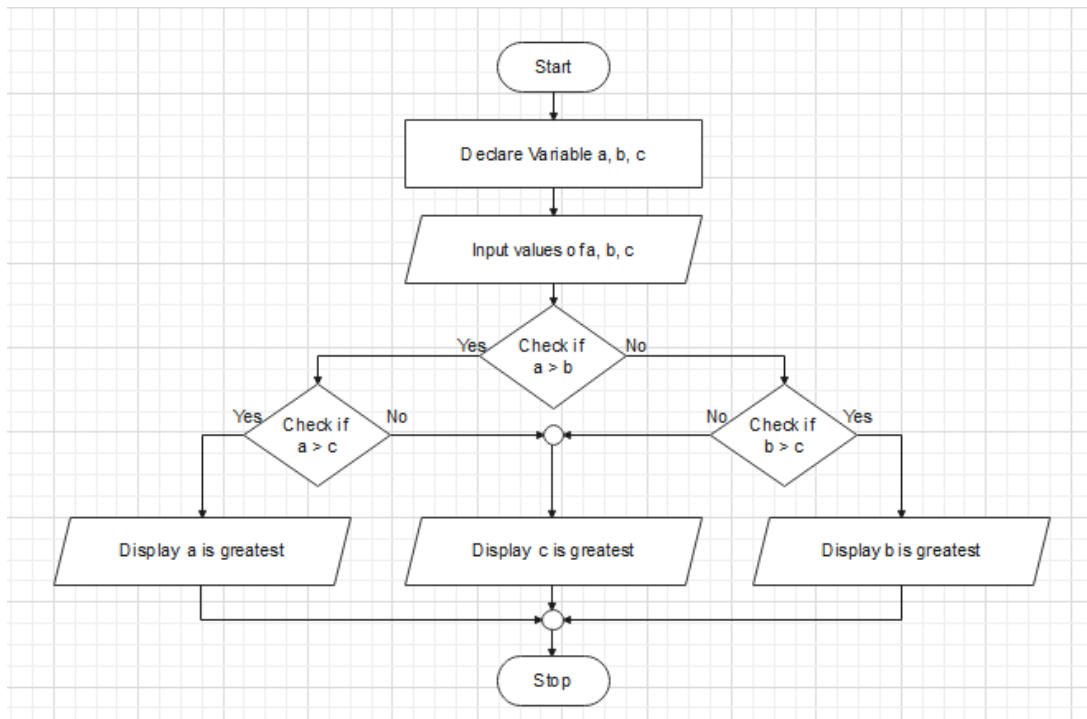
            Step 5.1.1.2: If Step 5.1.1 is false

                Step 5.1.1.2.1: Display c is greatest

Step 6: Stop



## Flowchart:



## Program:

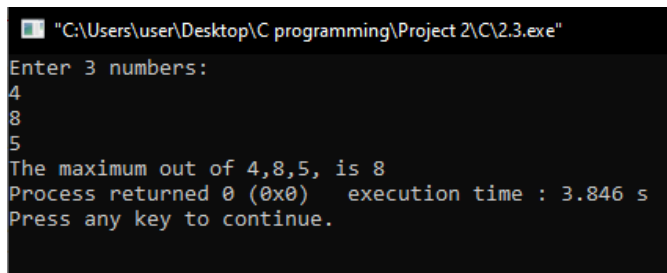
```
//Drishva Neupane 201648

//Question No 3

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

int main() {
    int a,b,c,Maximum;
    printf("Enter 3 numbers:\n");
    scanf("%d%d%d",&a,&b,&c);
    //Conditional operator is embedded
    Maximum = (a>b)?((a>c)?a:c):((b>c)?b:c);
    /*This expression can be read as if a is greater than b, then is a greater than c? if it is
    a is maximum and if not c is maximum. But if b is greater than a, is b greater than c? if it is
    b is greater and if nor c is greater*/
    printf("The maximum out of %d,%d,%d, is %d",a,b,c,Maximum);
    return 0;
}
```

## Output:



```
"C:\Users\user\Desktop\C programming\Project 2\C\2.3.exe"
Enter 3 numbers:
4
8
5
The maximum out of 4,8,5, is 8
Process returned 0 (0x0) execution time : 3.846 s
Press any key to continue.
```

## **4) Write a program to find out the real and equal roots of a quadratic equation using conditional operator.**

### Algorithm:

Step 1: Start

Step 2: Declare variable a, b, c

Step 3: Display "Input the coefficient of  $x^2$ , x and constant of the quadratic equation"

Step 4: Input coefficient of  $x^2$ , x and constant (i.e. a, b, c)

Step 5: Check if  $(b^2) - (4*a*c) > 0$

Step 5.1: If Step 5 is true

Step 5.1.1: Check if  $(b^2) - (4*a*c) == 0$

Step 5.1.1.1: If Step 5.1.1 is true

Step 5.1.1.1.1: The roots are equal,  $\text{Alpha} = \text{Beta} = -b / (2*a)$

Step 5.1.1.1.2: Display Alpha, Beta

Step 5.1.1.2: If Step 5.1.1 is false

Step 5.1.1.2.1:  $\text{Alpha} = (-b - ((b^2) - 4*a*c)^{0.5}) / (2*a)$

$\text{Beta} = (-b + ((b^2) - 4*a*c)^{0.5}) / (2*a)$

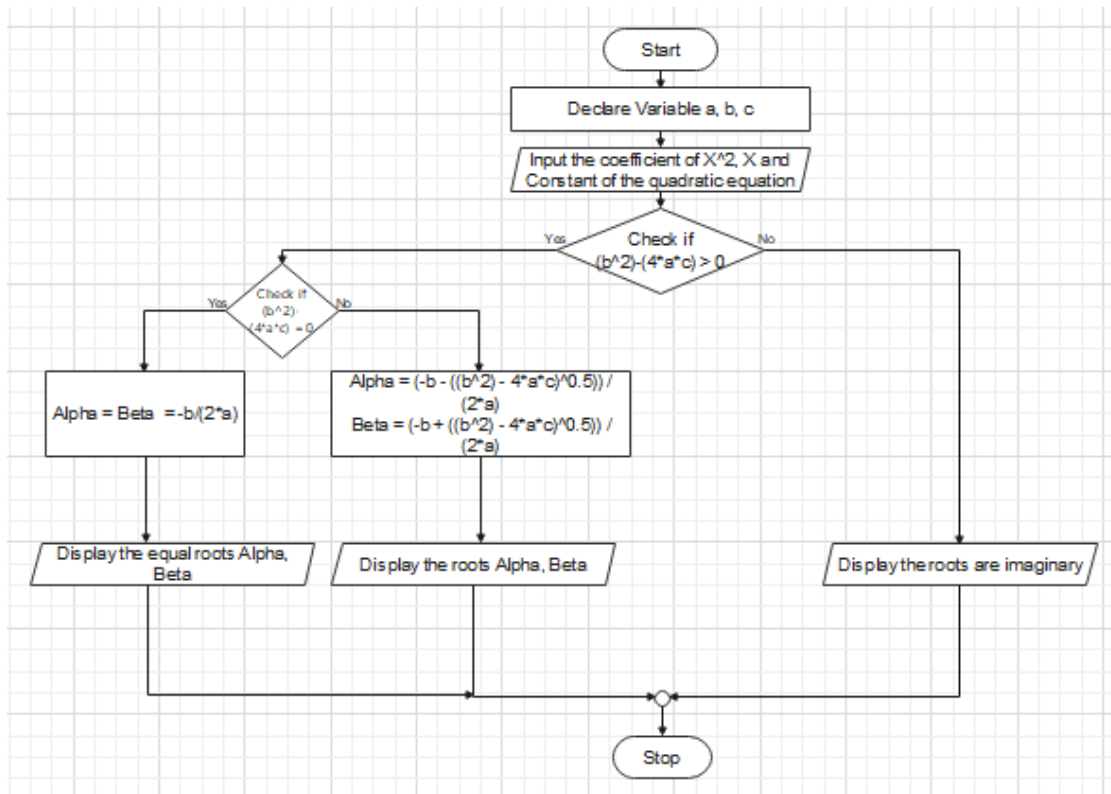
Step 5.1.1.2.2: Display Alpha, Beta

Step 5.2: If Step 5 is false

Step 5.2.1: Display the roots are imaginary

Step 6: Stop

## Flowchart:



## Program:

```
//Drishva Neupane 201648
//Program No.11

#include<stdio.h>
#include<math.h>
int main(){
    float a, b, c, value, alpha, beta;
    printf("Enter the coefficient of X^2, X and Constant of the quadratic equation whose roots are Alpha and Beta:\n");
    scanf("%f%f%f",&a,&b,&c);
    //B^2-4ac is used to check whether the roots are real imaginary or equal
    value = (pow(b,2) - (4*a*c));
    //if.....else if condition is used to check different conditions
    if(value > 0)//The roots are real
    {
        alpha = (-b + sqrt(value))/(2*a);
        beta = (-b - sqrt(value))/(2*a);
        printf("The roots are %.2f & %.2f",alpha,beta);
    }
    else if (value==0)//The roots are equal
    {
        alpha = beta = -b / (2 * a);
        printf("The roots Alpha = Beta = %.2f",alpha);
    }
    else //If none conditions are satisfied,the roots are imaginary
    {
        printf("The roots Alpha and Beta are imaginary");
    }
    return 0;
}
```

## Output:

```
"C:\Users\user\Desktop\C programming\Program 11.exe"
Enter the coefficient of X^2, X and Constant of the quadratic equation whose roots are Alpha and Beta:
1
-1
-2
The roots are 2.00 & -1.00
Process returned 0 (0x0)   execution time : 2.902 s
Press any key to continue.
```

```
"C:\Users\user\Desktop\C programming\Program 11.exe"
Enter the coefficient of X^2, X and Constant of the quadratic equation whose roots are Alpha and Beta:
2
-1
1
The roots Alpha and Beta are imaginary
Process returned 0 (0x0)   execution time : 2.239 s
Press any key to continue.
```

```
"C:\Users\user\Desktop\C programming\Project 2\C\2.4.exe"
Enter the coefficient of X^2, X and Constant of the quadratic equation whose roots are Alpha and Beta:
1
-10
25
The roots Alpha = Beta = 5.00
Process returned 0 (0x0)   execution time : 4.217 s
Press any key to continue.
```

**5) .Write a program to illustrate the modulus operator in which second is given as an input and the program converts it to hours, minutes and seconds.**

Algorithm:

Step 1: Start

Step 2: Declare variable x, hour, minute, second

Step 3: Display "Input the time period to be converted"

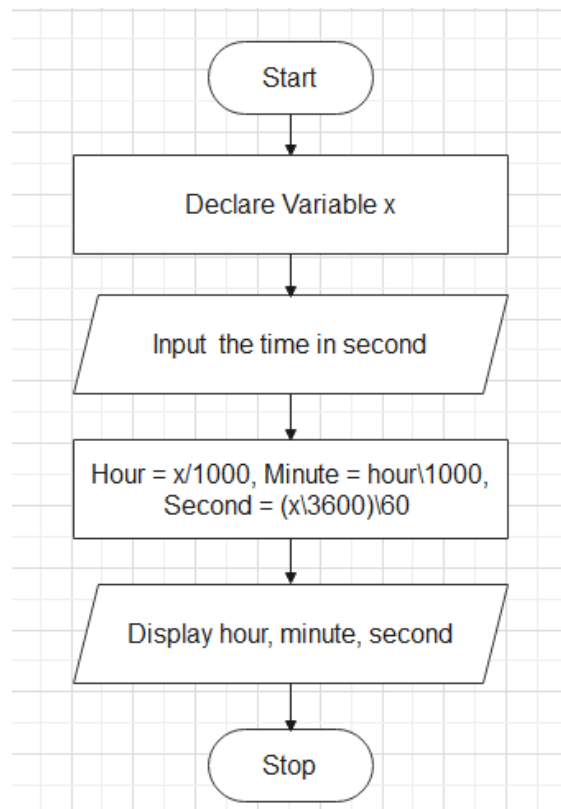
Step 4: Input the time in second

Step 5:  $\text{hour} = x/1000$ ,  $\text{minute} = \text{hour}\%1000$ ,  $\text{second} = (x\%3600)\%60$

Step 6: Display hour, minute, second

Step 7: Stop

Flowchart:



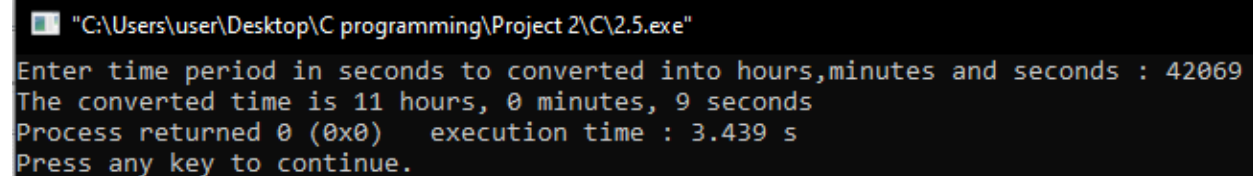
## Program:

```
//Drishva Neupane 201648

//Question no 5

#include<stdio.h>
int main() {
    int x, hour, minute, second;
    //Here the variables are taken in int so that modulus work properly
    printf("Enter time period in seconds to converted into hours, minutes and seconds : ");
    scanf("%d", &x);
    hour = x/3600;
    //Here dividing the second by 3600 gives exact hour as it is taken as int
    minute = hour/60;
    //Similarly, doing the same gives exact minute
    second = (x%3600)%60;
    //As we're performing operations in seconds, modulus is used in the value provided.
    //First modulus gives the remainder value in minute, the other converts it to second
    printf("The converted time is %d hours, %d minutes, %d seconds", hour, minute, second);
    return 0;
}
```

## Output:



```
"C:\Users\user\Desktop\C programming\Project 2\C\2.5.exe"
Enter time period in seconds to converted into hours, minutes and seconds : 42069
The converted time is 11 hours, 0 minutes, 9 seconds
Process returned 0 (0x0)   execution time : 3.439 s
Press any key to continue.
```

**6) Write a program to read the radius of circle and calculate the circumference and area**

**Algorithm:**

Step 1: Start

Step 2: Declare variable weight, height1, height2

Step 3: Display "Enter your height in feet and inches and your weight in kilogram"

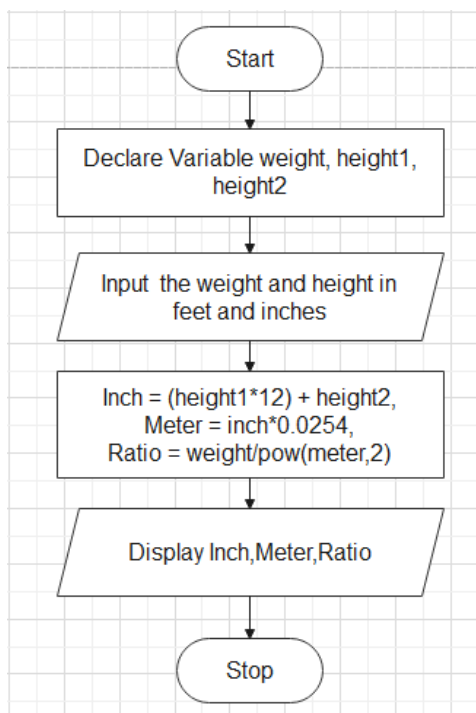
Step 4: Enter your height in feet (as height1) and inches as (height 2) and your weight in kilogram

Step 5:  $\text{inch} = (\text{height1} * 12) + \text{height2}$ ,  $\text{meter} = \text{inch} * 0.0254$ ,  $\text{ratio} = \text{weight} / \text{pow}(\text{meter}, 2)$

Step 6: Display inch, meter & ratio

Step 7: Stop

**Flowchart:**



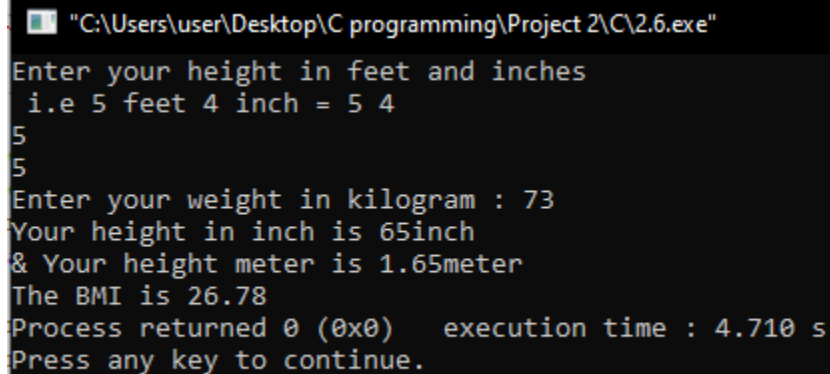
## Program:

```
//Drishya Neupane 201648

//Question no 5

#include<stdio.h>
#include<math.h>
int main() {
    float weight, meter, ratio;
    int feet, inch, height1, height2;
    printf("Enter your height in feet and inches\n i.e 5 feet 4 inch = 5 4\n");
    scanf("%d%d", &height1, &height2);
    printf("Enter your weight in kilogram : ");
    scanf("%f", &weight);
    inch = (height1*12) + height2;
    printf("Your height in inch is %dinch", inch);
    meter = inch*0.0254;
    printf("\n& Your height meter is %.2fmeter", meter);
    ratio = weight/pow(meter, 2);
    printf("\nThe BMI is %.2f", ratio);
    return 0;
}
```

## Output:



```
"C:\Users\user\Desktop\C programming\Project 2\C\2.6.exe"
Enter your height in feet and inches
i.e 5 feet 4 inch = 5 4
5
5
Enter your weight in kilogram : 73
Your height in inch is 65inch
& Your height meter is 1.65meter
The BMI is 26.78
Process returned 0 (0x0)   execution time : 4.710 s
Press any key to continue.
```