Programming in C

**Drishya Neupane** **Ms.Rabina Chaudhary**

BE Software Programming in C

Roll no: 201648 Software Department

**Practical No: 3**

**Title:** TO LEARN DECISION CONTROL STATEMENTS

**Objective:**

1. Understanding the conditional statements and decision control 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. **Write a program to check whether input character is lowercase character, uppercase character, digits or special symbols.**

Algorithm:

Step 1: Start

Step 2: Declare character cha

Step 3: Display "Input a character"

Step 4: Input the value of character cha

Step 5: Check cha >= A and cha <= Z

Step 5.1: If Step 5 is true

Step 5.1.1: Display the character is uppercase

Step 5.2: If Step 5 is false

Step 5.2.1: Check if cha >= 'a' and cha <= 'z'

Step 5.1.1.1: If Step 5.2.1 is true

Step 5.1.1.1.1: Display the character is lowercase

Step 5.1.1.2: If Step 5.1.1 is also false

Step 5.1.1.2.1: Check if cha >= '0' && cha <= '9'

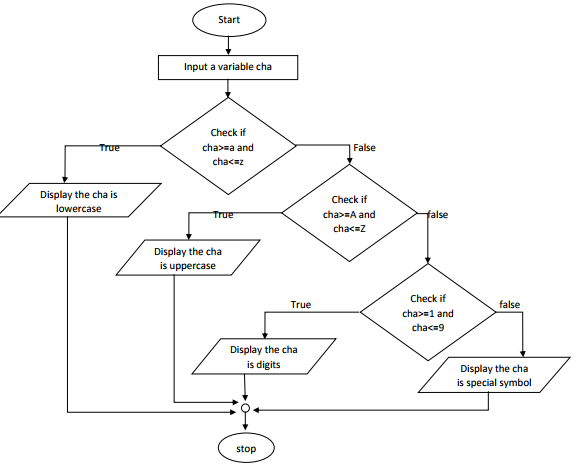
Step 5.1.1.2.1.1: Display the character is numeric

Step 5.1.1.2.1.2: If Step 5.1.1.2.1 is also false

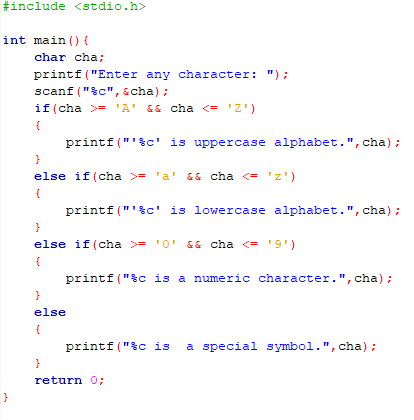
Step 5.1.1.2.1.2.1: Display the character is special symbol

Step 6: Stop

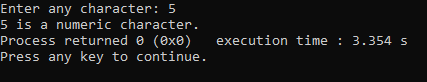
Flowchart:



Program:



Output:



1. **Write a program to input a number and display which day of the week on the basis of input number.**

Algorithm:

Step 1: Start

Step 2: Declare int a

Step 3: Display "Input a number betw 1 to 7"

Step 4: Input the value of number a

Step 5: Check a = 1

Step 5.1: If Step 5 is true

Step 5.1.1: Display the day is Sunday

Step 5.2: If Step 5 is false, check if a = 2

Step 5.2.1: Display the day is Monday

Step 5.3: If Step 5 is false, check if a = 3

Step 5.1.1: Display the day is Tuesday

Step 5.4: If Step 5 is false, check if a = 4

Step 5.2.1: Display the day is Wednesday

Step 5.5: If Step 5 is false, check if a = 5

Step 5.1.1: Display the day is Thursday

Step 5.6: If Step 5 is false, check if a = 6

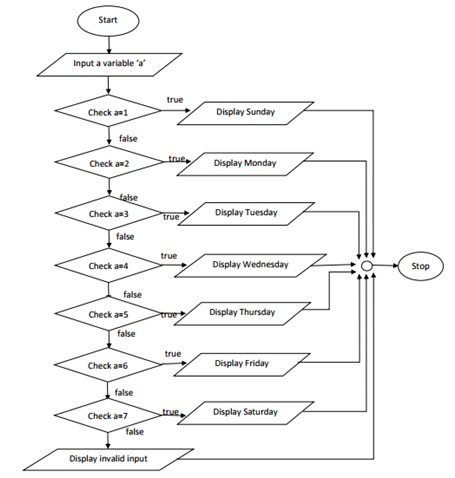
Step 5.2.1: Display the day is Friday

Step 5.7: If Step 5 is false, check if a = 7

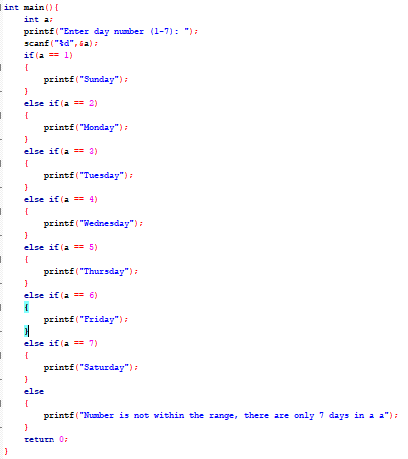
Step 5.1.1: Display the day is Saturday

Step 6: Stop

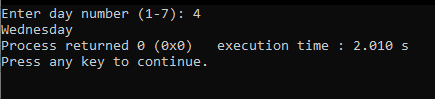
Flowchart:



Program:



Output:



1. **Write a program to find the number given by user is exactly divisible by 2, 3 and 6 or not.**

Algorithm:

Step 1: Start

Step 2: Declare int num

Step 3: Display "Input a number"

Step 4: Input the value of number num

Step 5: Check if (num \ 2 == 0) and (num \ 3 == 0) and (num \ 6 == 0)

Step 5.1: If Step 5 is true

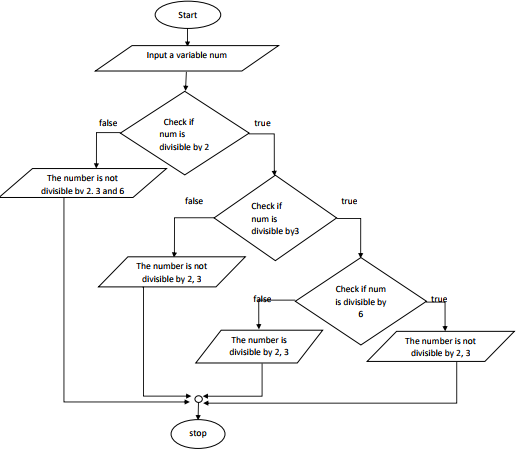
Step 5.1.1: Display the number is exactly divisible by 2,3,6

Step 5.2: If Step 5 is false

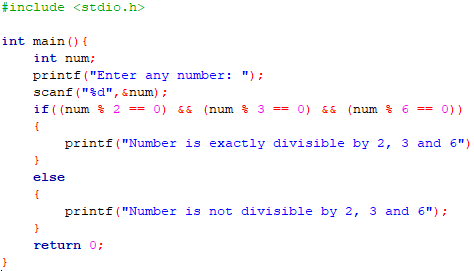
Step 5.2.1: Display the number is not divisible by 2,3,6

Step 6: Stop

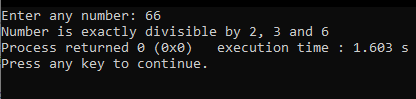
Flowchart:



Program:



Output:



1. **Write a program to find maximum of 3 numbers using else if ladder.**

Algorithm:

Step 1: Start

Step 2: Declare int a, b, c

Step 3: Display "Input three number"

Step 4: Input the value of three numbers

Step 5: Check if a => b and a => c

Step 5.1: If Step 5 is true, Display a is the greatest

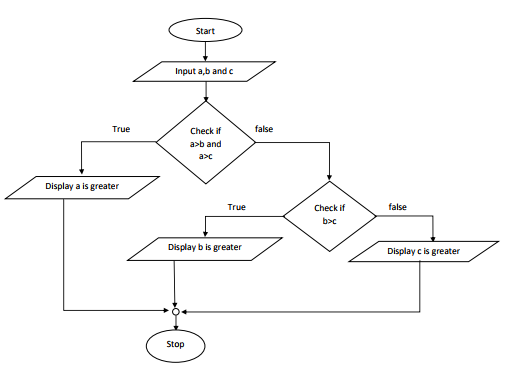
Step 5.2: If Step 5 is false, Check if b => a and b => c

Step 5.2.1: If Step 5.2 is true, Display b is the greatest

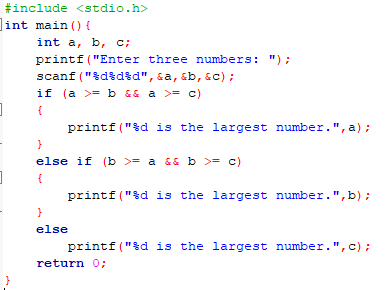
Step 5.3: If Step 5.1 & 5.2 is false, Display c is the greatest

Step 6: Stop

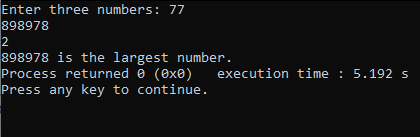
Flowchart:



Program:



Output:



1. **Write a program to find maximum of 4 numbers using else if ladder**

Algorithm:

Step 1: Start

Step 2: Declare int a, b, c, d

Step 3: Display "Input four number"

Step 4: Input the value of four numbers

Step 5: Check if a => b and a => c and a=>d

Step 5.1: If Step 5 is true, Display a is the greatest

Step 5.2: If Step 5 is false, Check if b => a and b => c and b=>d

Step 5.2.1: If Step 5.2 is true, Display b is the greatest

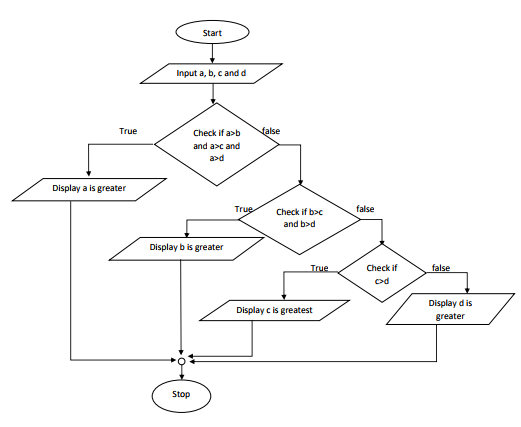
Step 5.3: If Step 5 is false, Check if c => a and c => b and c=>d

Step 5.2.1: If Step 5.3 is true, Display c is the greatest

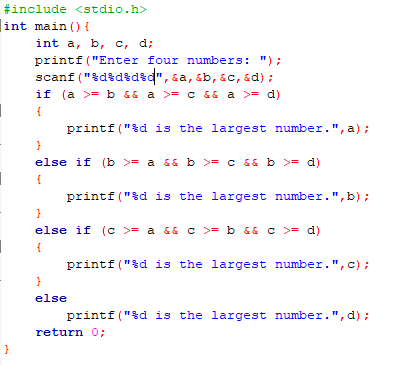
Step 5.4: If Step 5.1,5.2 & 5.3 is false, Display d is the greatest

Step 6: Stop

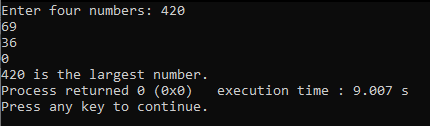
Flowchart:



Program:



Output:



1. **Write a program to find maximum of 3 numbers using nested if else.**

Algorithm:

Step 1: Start

Step 2: Declare int a, b, c

Step 3: Display "Input three number"

Step 4: Input the value of three numbers

Step 5: Check if a > b

Step 5.1: If Step 5 is true, check if a>c

Step 5.1.1: If Step 5.1 is true, Display a is greatest

Step 5.1.2: If Step 5.1 is false, Display c is greatest

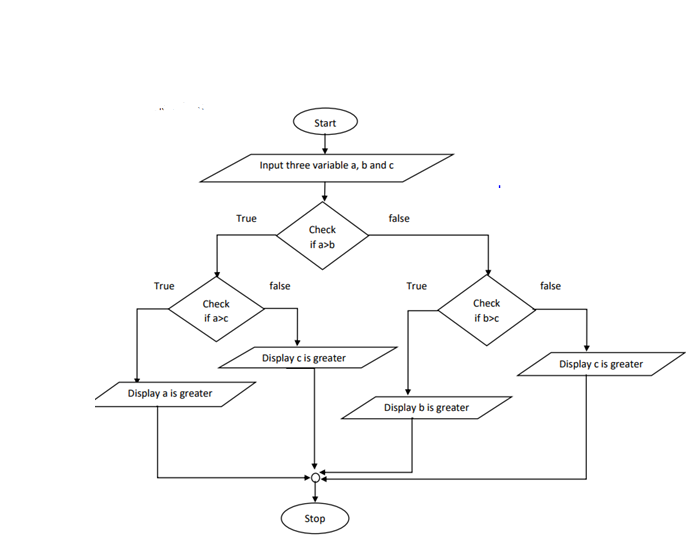
Step 5.2.1: If Step 5 is false, check if b > c

Step 5.2.1: If Step 5.1 is true, Display b is greatest

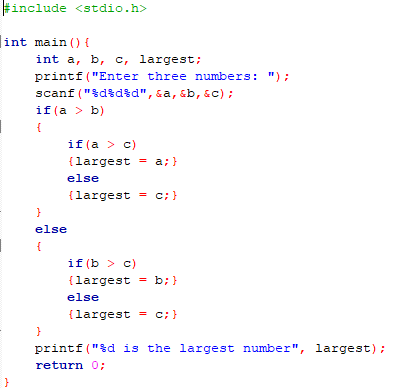
Step 5.2.2: If Step 5.1 is false, Display c is greatest

Step 6: Stop

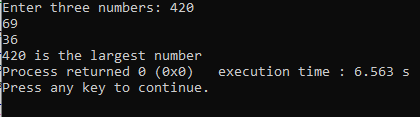
Flowchart:



Program:



Output:



1. **Write a program to find maximum of 4 numbers using nested if else.**

Algorithm:

Step 1: Start

Step 2: Declare int a, b, c, d

Step 3: Display "Input four number"

Step 4: Input the value of four numbers

Step 5: Check if a > b

Step 5.1: If Step 5 is true, check if a>c

Step 5.1.1: If Step 5.1 is true, check if a > d

Step 5.1.1.1: If Step 5.1.1 is true, Display a is greatest

Step 5.1.1.2: If Step 5.1.1 is false, Display d is greatest

Step 5.1.2: If Step 5 is false, check if c > d

Step 5.1.2.1: If Step 5.1.2 is true, Display c is greatest

Step 5.1.2.2: If Step 5.1.2 is false, Display d is greatest

Step 5.2: If Step 5 is false, check if b>c

Step 5.1.1: If Step 5.1 is true, check if b > d

Step 5.1.1.1: If Step 5.1.1 is true, Display b is greatest

Step 5.1.1.2: If Step 5.1.1 is false, Display d is greatest

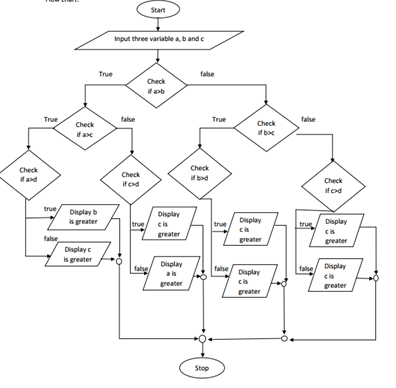
Step 5.1.2: If Step 5 is false, check if c > d

Step 5.1.2.1: If Step 5.1.2 is true, Display c is greatest

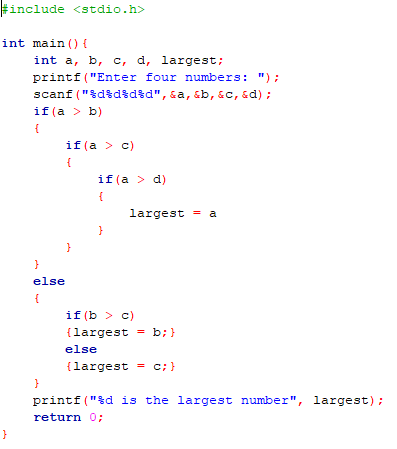
Step 5.1.2.2: If Step 5.1.2 is false, Display d is greatest

Step 6: Stop

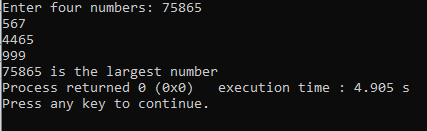
Flowchart:



Program:



Output:



1. **An electricity board charges according to the following rate**

**i. For the first 100 units -------- Rs 40 per unit**

**ii. For the next 200 units -------- Rs 50 per unit**

**iii. For beyond 300 units --------- Rs 60 per unit**

**All users are also charged meter charge, which is equal to Rs. 50. Write a program to read number of units consumed and print the total charge.**

Algorithm:

Step 1: Start

Step 2: Declare int unit

Step 3: Display "Input units consumed"

Step 4: Input the unit consumed

Step 5: Check if unit > 100

Step 5.1: If Step 5 is true, Charge = 40 \* unit

Step 5.2: If Step 5.1 is false, check if unit > 100 and unit <= 300

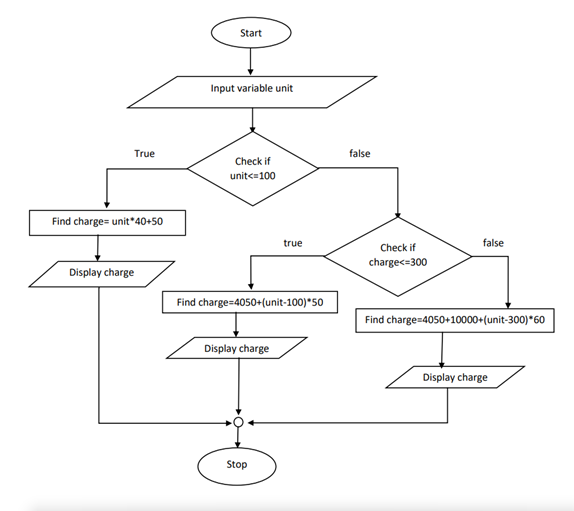
Step 5.2.1: If Step 5.2 is true, charge = 4000 + ((units - 100) \* 50)

Step 5.3: If Step 5.2 is false, charge = 14000 + ((units - 300) \* 60)

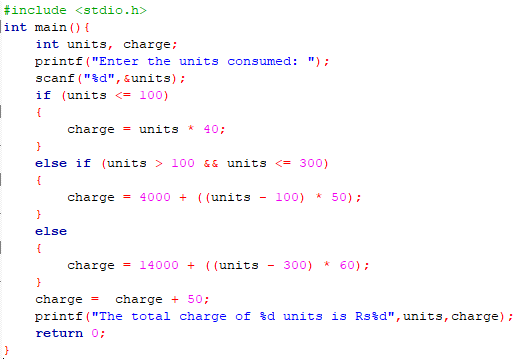
Step 6: Display charge

Step 7: Stop

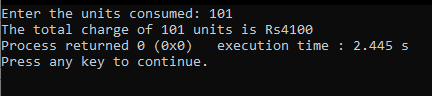
Flowchart:



Program:



Output:



1. **Write a program that reads marks of student in seven subjects. Calculate the percentage if the student has achieved greater than 45 in each subject and use these conditions**

**i. Percentage greater than equals to 80 is distinction**

**ii. Percentage from 60 to 79 is first division**

**iii. Percentage from 45 to 59 is second division**

Algorithm:

Step 1: Start

Step 2: Declare int a,b,c,d,e,f,g;

Step 3: Display "Input marks obtained in all 7 subjects"

Step 4: Input the marks obtained in all 7 subjects

Step 5: Check if a >= 45 & b >= 45 & c >= 45 & d >= 45 & e >= 45 & f >= 45 & g >= 45

Step 5.1: If Step 5 is true, percentage = ((a+b+c+d+e+f+g)/700)\*100

Step 5.1.1: Check if percentage >= 80

Step 5.1.1.1: If Step 5.1.1 is true, display Distinction

Step 5.1.2: If Step 5.1.1 is false, Check if 60 <= percentage <= 79

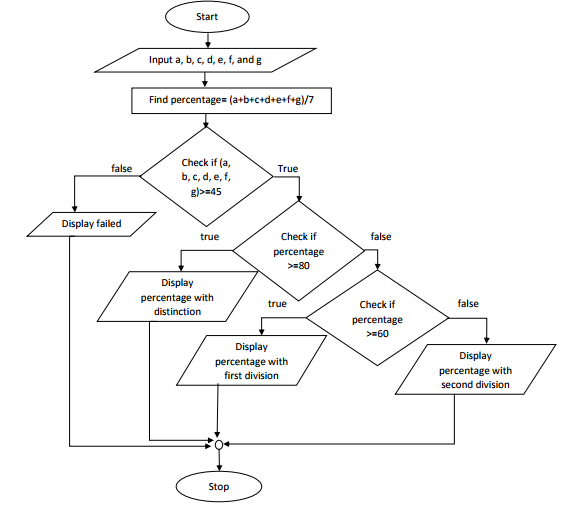
Step 5.1.2.1: If Step 5.1.1 is true, display First Division

Step 5.1.3: If Step 5.1.2 is false, display Second Division

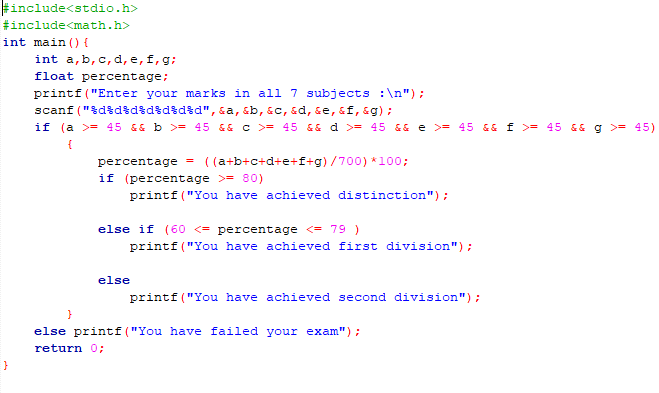
Step 5.2: If Step 5 is false, Display Failed

Step 6: Stop

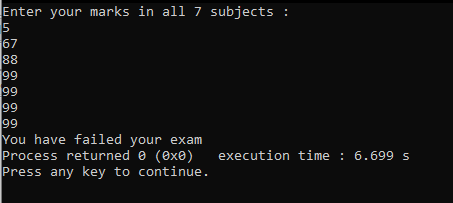
Flowchart:



Program:



Output:



1. **An electricity board charges according to the following rates**

**i. For the first 20 units -------- Rs 80**

**ii. For the next 80 units -------- Rs 7.5 per unit**

**iii. For next 100 units ----------- Rs 8.5 per unit**

**iv. For beyond 200 units ------- Rs 9.5 per unit**

**All users are also charged meter charge, which is equal to Rs. 50. Also tax = 5% of charge. Write a program to read number of units consumed and print the total charge.**

Algorithm:

Step 1: Start

Step 2: Declare int unit

Step 3: Display "Input units consumed"

Step 4: Input the unit consumed

Step 5: Check if units <= 20

Step 5.1: If Step 5 is true, Charge = 80 \* unit

Step 5.2: If Step 5.1 is false, check if units > 20 && units <= 100

Step 5.2.1: If Step 5.2 is true, charge = 1600 + ((units - 20) \* 7.5)

Step 5.3: If Step 5.2 is false, check if nits > 100 && units <= 200

Step 5.3.1: If Step 5.3 is true, charge 2200 + ((units - 100) \* 8.5)

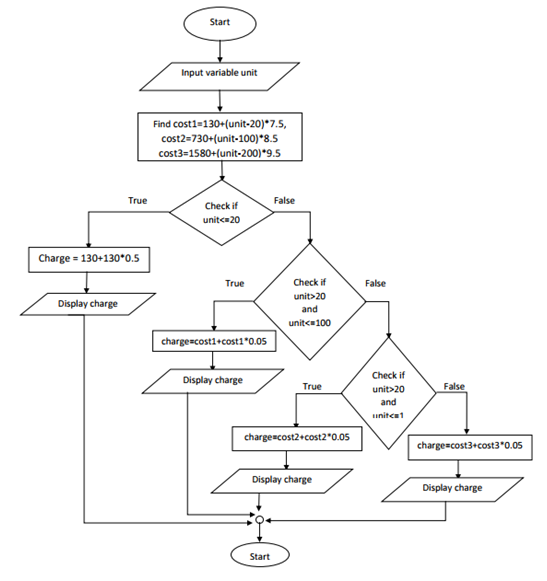
Step 5.4: If Step 5.3 is false, charge = 3050 + ((units - 200) \* 9.5)

Step 6: Charge = (charge + 50) + 0.5 \* (charge + 50)

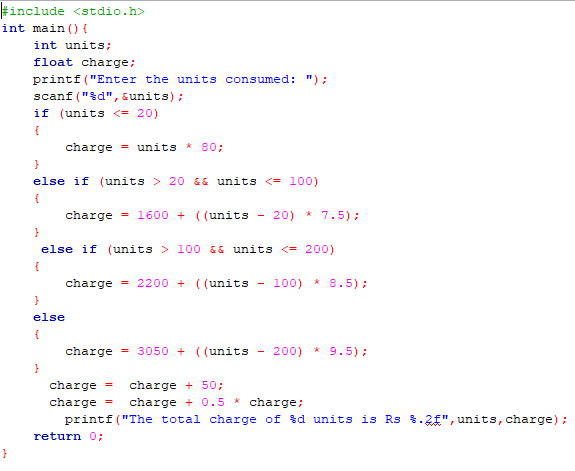
Step 7: Display charge

Step 8: Stop

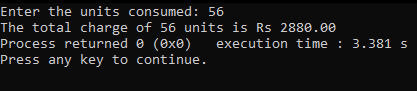
Flowchart:



Program:



Output:



1. **A company does insurance of the driver in following cases**

**a. If the driver is married**

**b. If the driver is unmarried male and above 30 years of age**

**c. If the driver is unmarried female and above 25 years of age  Write the program without using any logical operators**

Algorithm:

Step 1: Start

Step 2: Declare int age; character status, gender

Step 3: Display "Input Y if you're married, if not enter N" and "Input M if you're male & F for female and age"

Step 4: Input the Y or N, M or F and age

Step 5: Check if Status = Y

Step 5.1: Display Eligible for Insurance

Step 6: Check Gender = M

Step 6.1: If Step 6 is true, check if age > 30

Step 6.1.1: If Step 6.1 is true, display Eligible for Insurance

Step 6.1.2: If Step 6.1 is false, display Not Eligible for Insurance

Step 6.2: If Step 6 is false, check if gender = F

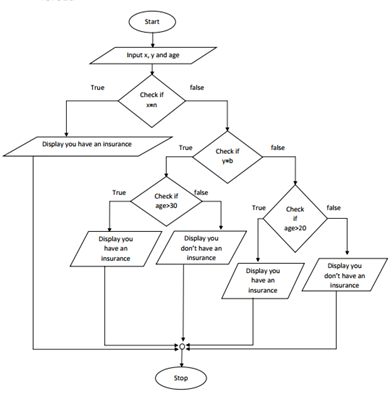
Step 6.2.1: If Step 6.2 is true, check if age > 25

Step 6.2.1.1: If Step 6.2.1 ia true, display Eligible for Insurance

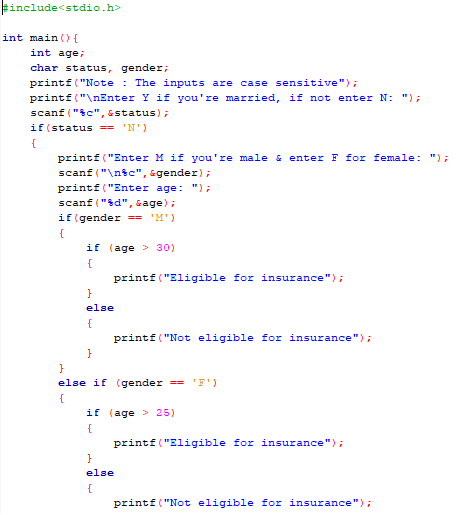
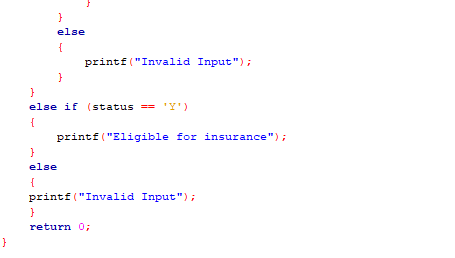
Step 6.2.1.2: If Step 6.2.1 is false, display Not Eligible for Insurance

Step 6: Stop

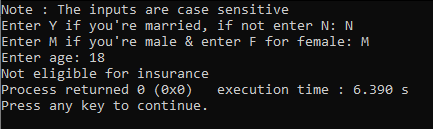
Flowchart:



Program:

Output:



1. **A company does insurance of the driver in following cases**

**a. If the driver is married**

**b. If the driver is unmarried male and above 30 years of age**

**c. If the driver is unmarried female and above 25 years of age  Write the program using logical operators**

Algorithm:

Step 1: Start

Step 2: Declare int age; character status, gender

Step 3: Display "Input Y if you're married, if not enter N"

Step 4: Input the Y or N

Step 5: Check if Status = Y

Step 5.1: Display Eligible for Insurance

Step 6: If Status = N

Step 6.1: Display "Input M if you're male & F for female and age"

Step 6.1.1: Check if gender = M and age > 30

Step 6.1.1.1: If Step 6.1.1 is true, display Eligible for Insurance

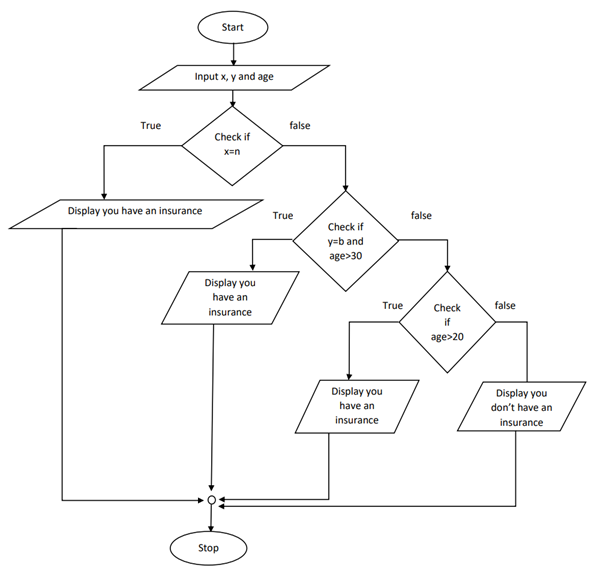
Step 6.1.2: Check if gender = F and age > 25

Step 6.1.2.1: If Step 6.1.2 is true, display Eligible for Insurance

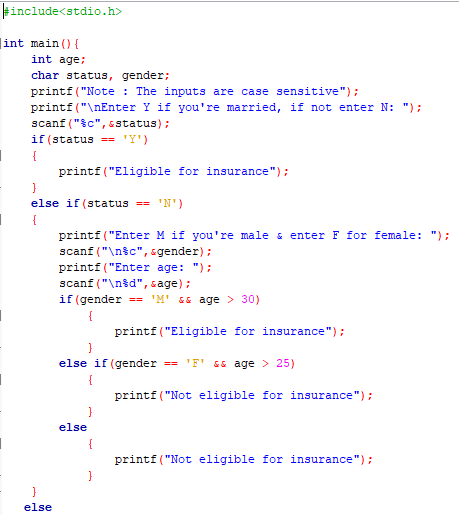
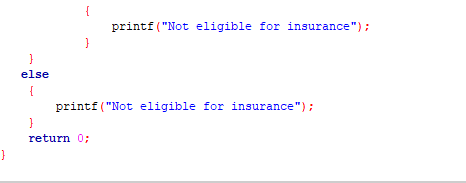
Step 5.2: If Step 6.1.1 and 6.1.2 is false, Display Not Eligible for Insurance

Step 7: Stop

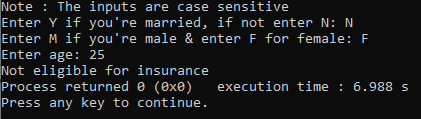
Flowchart:



Program:

Output:



1. **Write a program to read a character from keyboard and convert the input**

**Into uppercase if it is lowercase and vice versa.**

Algorithm:

Step 1: Start

Step 2: Declare char ch

Step 3: Display "Input your character"

Step 4: Input character

Step 5: Check if ch >= 'A' && ch <= 'Z'

Step 5.1: If Step 5 is true, lowercase = ch - 32

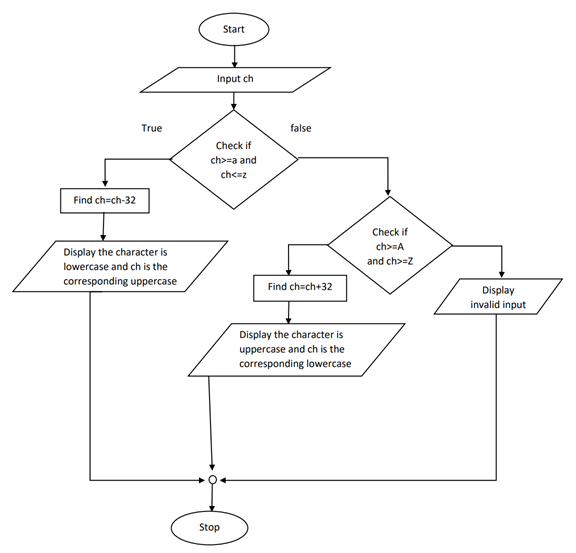
Step 5.1.1: Print equivalent lowercase of ch

Step 5.2: If Step 5 is true, lowercase = ch + 32

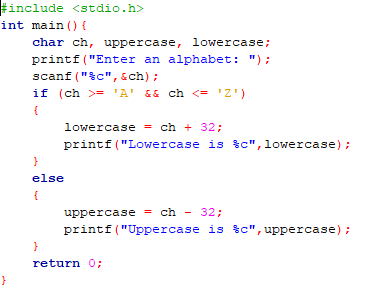
Step 5.2.1: Print equivalent uppercase of ch

Step 6: Stop

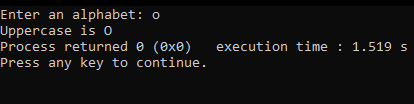
Flowchart:



Program:



Output:



1. **Write a program to swap two input numbers.**

Algorithm:

Step 1: Start

Step 2: Declare int Number1,Number2

Step 3: Display "Input two numbers"

Step 4: Input two numbers

Step 5: Number1 = Number1 + Number2

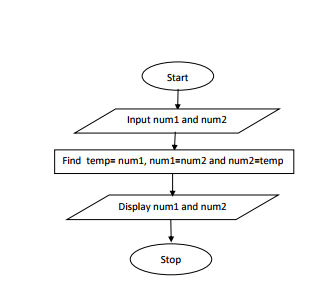
Step 6: Number2 = Number1 - Number2

Step 7: Number1 = Number1 - Number2

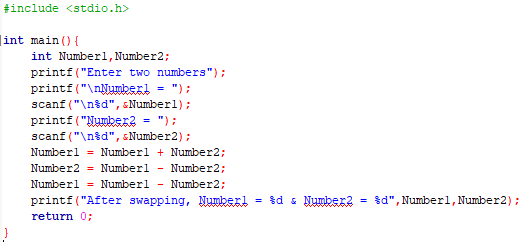
Step 8: Display Number1 and Number2

Step 9: Stop

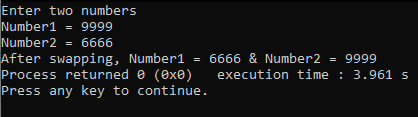
Flowchart:



Program:



Output:



1. **A year is entered through the keyboard, write a program to determine whether it is leap year or not**

Algorithm:

Step 1: Start

Step 2: Declare int year

Step 3: Display "Input a year"

Step 4: Input year

Step 5: Check if year\400 = 0

Step 5.1: If Step 5 is true, display it is a leap year

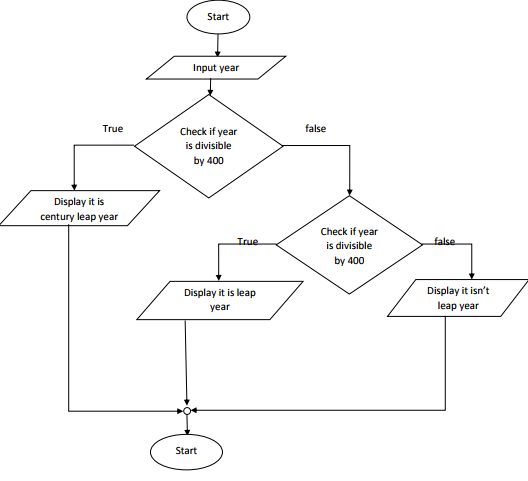
Step 5.2: If Step 5 is false, check if year\4 = 0

Step 5.2.1: If Step 5.2 is true, display it is a leap year

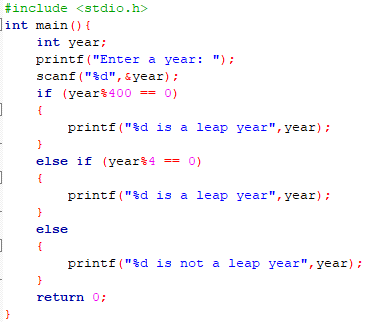
Step 5.3: If Step 5.1 and 5.2 is false, display it isn't a leap year

Step 6: Stop

Flowchart:



Program:



Output:

