Date. No.

(X) Control Structures:

Control structures are the statements than control the flow of the program. Control structures are branching and looping statements.

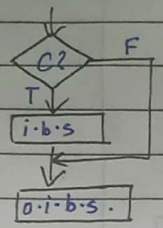
A) Branching statement:

Branching statement change the path of execution of the program based upon given conditions.

They are: if, if else, else-if, nested if-else, switch.

a): if - statement:

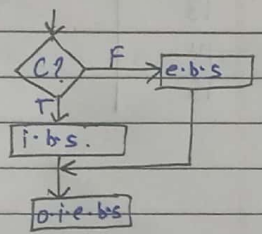
Flowchart:



Syntax:
```
if (condition)
{ if block statements; }
outside if block statements;
```

b): if-else statement:

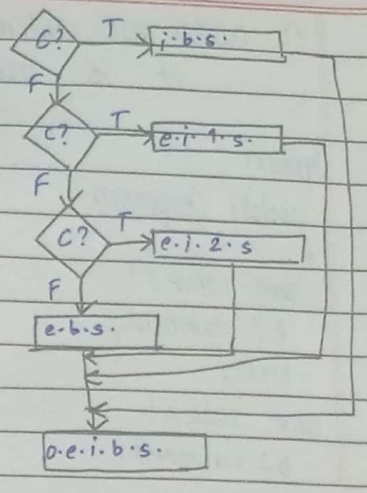
Flowchart:



Syntax:
```
if (condition)
{ if block statement; }
else
{ else block statement; }
outside if-else statements.
```

c) else-if statement:

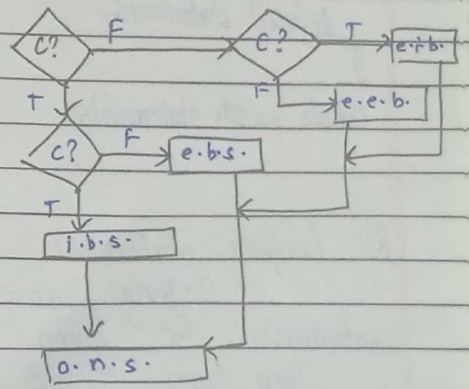
Flowchart:



Syntax:
```
if (condition)
{ if-block statements; }
else if (condition)
{ else if block 1 statement; }
else if (condition)
{ else if block 2 statement; }
⋮
else
{ else block statement; }
outside else if block statement.
```

(d) Nested if-else:

Flowchart:



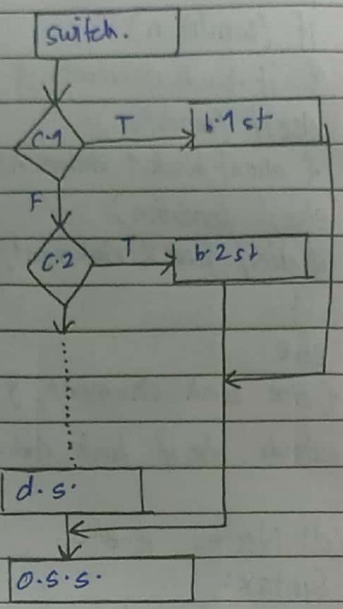
Syntax:
```
if (condition)
{
  if (condition)
  { if block statement; }
  else
  { else block statement; }
}
else.
{
  if (condition)
  { if-block statement; }
  else
  { else-block statements; }
}
outside next statement;
```

(e): switch-case statement:

It is used for menu-based programs

Syntax:

```
switch (expression)
{
 case value1:
  b1 statements;
  break;
 case value2:
  b2 statements;
  break;
 .
 .
 .
 default
  default statements;
}
outside switch statements;
```



(B): Looping statements:

Looping statements repeats a set of statements for given number of times.

They are: for loop, while loop, do-while loop, range-based loop.

a) for loop:

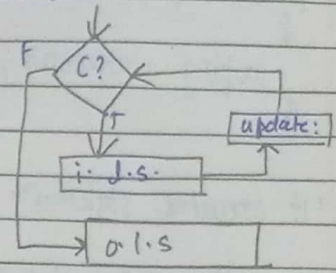
→ It is entry-controlled loop

→ Number of iterations are known.

Syntax:

```
for (initialization; condition; update)
{
   looping statements;
}
outside loop statements;
```

Flowchart:



b) while loop:

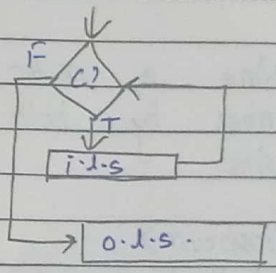
→ It is entry-controlled loop.

→ Number of iterations are not known.

Syntax:

```
while (condition)
{
   looping statements;
}
outside loop statements;
```

Flowchart:

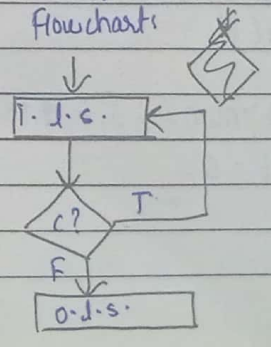


(c): do loop: It is exit-controlled loop executed atleast once.

Syntax:

```
do
{ looping statements; }
while (condition);
outside loop statements;
```

Flowchart:



(d) Range based for loop:

Syntax:

```
for (range declaration : range expression)
{
  looping statements;
}
```

* It executes statement for each range expression.

Eg:
```
int a[] = {1,2,3}.
for (int x : a)
{
  cout << x << endl;
}
```

*) Output:

1
2
3

<Q>: Define a vector and check divisibility of a numbers by both 3 and 5.

Ans:

```
#include <iostream>
#include <vector> // used to declare vector.
using namespace std;
void main()
{
vector <int> num = {3,6,9,15,17,18,21,55,100,200,300};
int count = 0;
for (auto a:num)
{
  if (n%3==0 && n%5==0)
  { count++; }
}
cout << count; << "numbers divisible" << endl;
}
```

*) Output

2

2 numbers divisible