Control Structure:

Control structure is a way to specify the flow of ~~co~~ control in programs.

It analyzes and chooses in which direction a program flows based on certain parameters and conditions.

There are as follows:

(i) Conditional statements.
(ii) Loop statements
(iii) Break control statements.

<A>: Conditional statement:

Condition statements are used to make decision based on given conditions and draw conclusion.

In C programming, condition ~~co~~ statements are ~~g~~ ~~divided into~~ of three types:

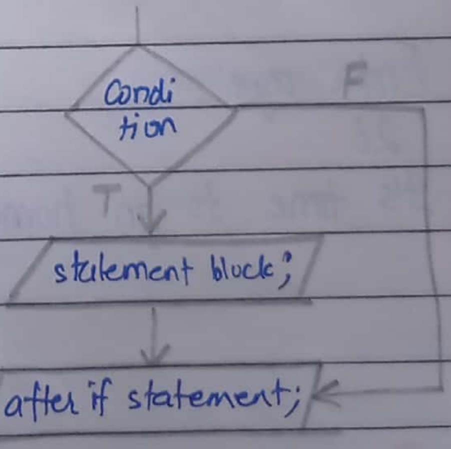
i) if statement
ii) if-else statement
iii) switch statement
iv) if-else-if ladder
v) nested if.

i) if statement:

*) Syntax:

```
 if (condition)
        statement;
after if statements.
```

*) Flowchart



Date. No.

Note: i) if single statement after ~~if~~ if, no need to use curly braces
ii) if multiple statement block, put curly braces.

iii) if (condition);
this semicolon terminates the if call and rested of the ~~sentences~~ statements is printed.

Eg: ⓧ
```
# include <stdio.h>
{
int age;
printf("Enter age: \n");
scanf("%d", &age);
if (age == 25)
{
  printf ("Your age is = %d", age);
  printf ("\n You can go coeffee with me\n");
}
printf ("It's time to go home");
}
```

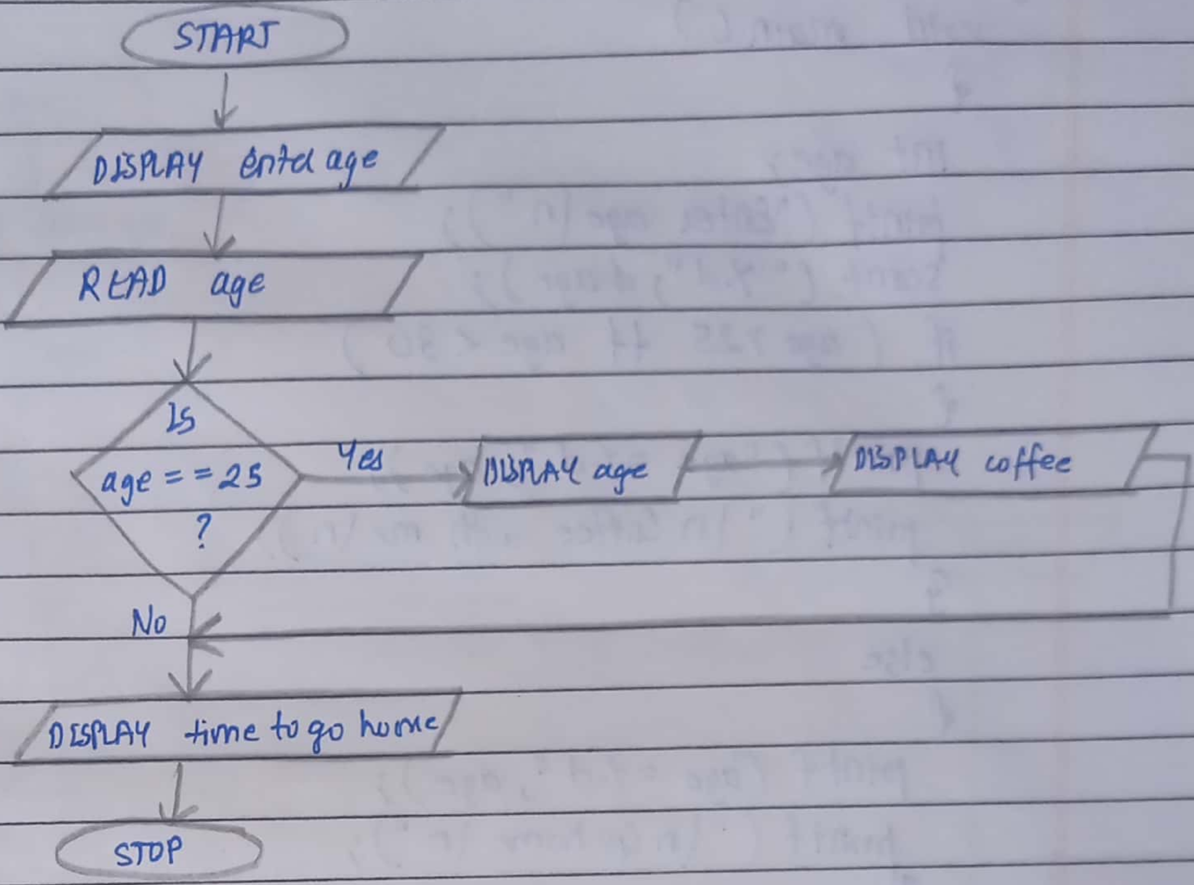
ⓧ Outputs:

```
Enter age                              Enter age
 26                                    25
It's time to go home                   Your age is 25
                                       You can go coeffee with me.
                                       It's time to go home.
```

पाठशाला

Date.   No.

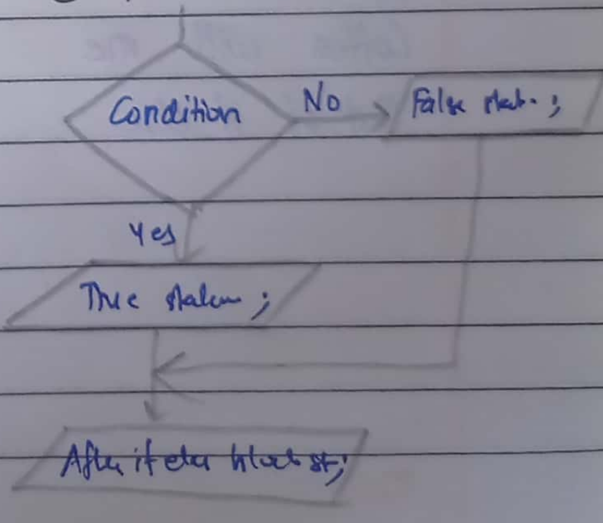
⊛ Flowchart:



ii) if - else statement:
→ It is the extension of simple if statement. Here, we have two blocks ie, true block statement and false block statements.

⊛ Syntax
```
if (condition)
{
  true block statement; }
else
{
  false block statement; }
after if - else block statement.
```

⊛ Flow chart:



Eg:
```
# include <stdio.h>
void main ()
{
 int age;
 printf ("Enter age \n ");
 scanf (" %d", & age);
if ( age >25 && age < 30 )
{
 printf ("age =%d", age);
 printf (" \n Coffee with me \n );
}
else
{
 printf ("age =%d ", age);
 printf (" \n Go home \n ");
}
printf ("Out of if-else");
}
```

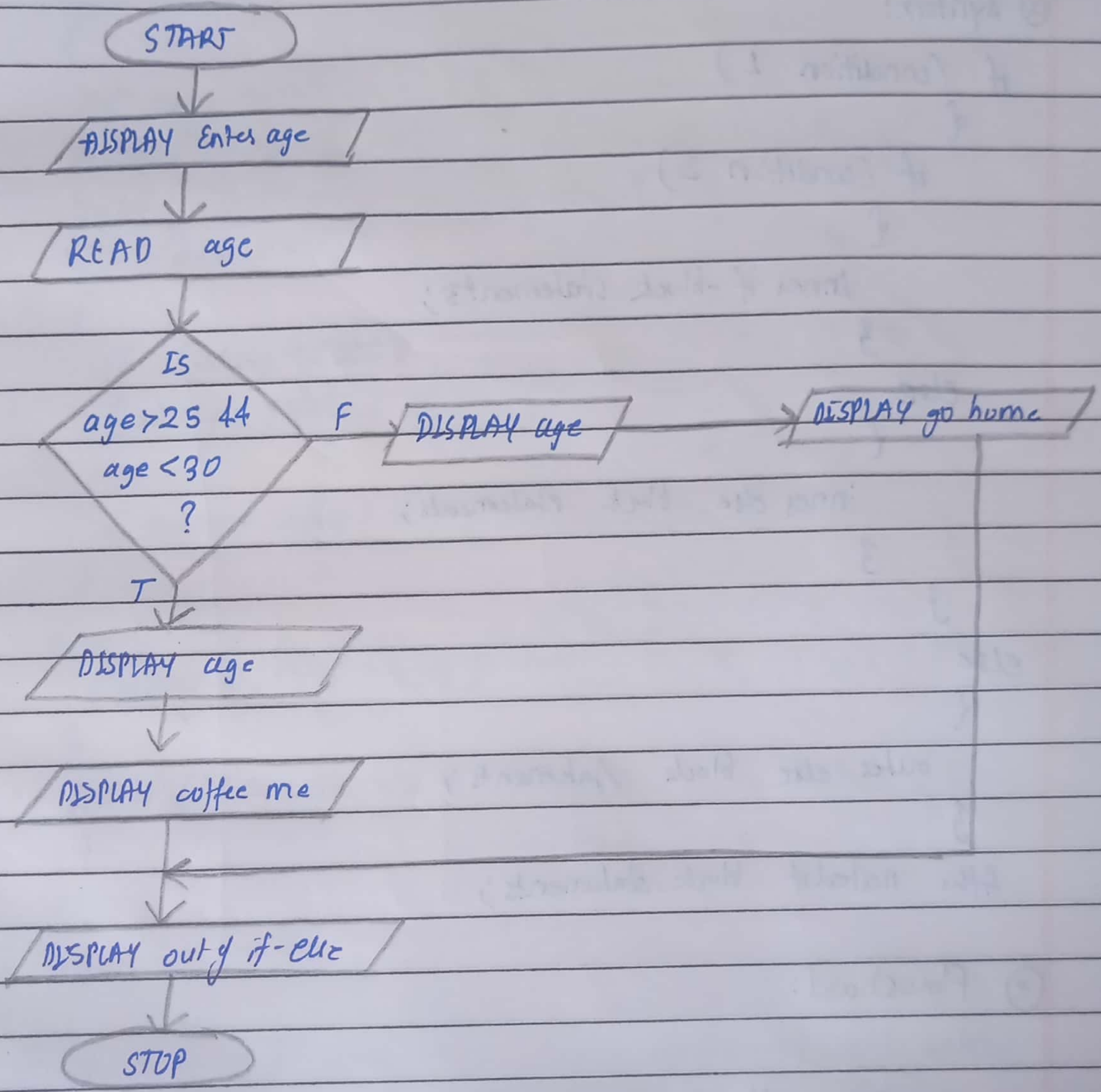
*) Outputs:

```
Enter age
28
Coffee with me
Out of if-else
```

```
Enter age
31
Go Home
Out of if-else.
```

Date. No.

*) Flowchart:

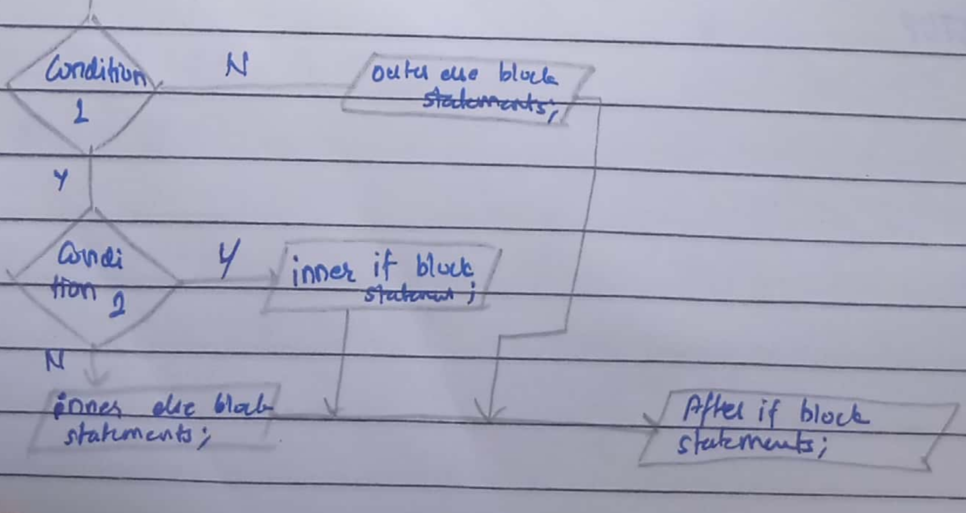


(iii) Nested-if

④ Syntax:

```
if (condition 1)
  {
     if (condition 2)
       {
          Inner if -block statements;
       }
    else
       {
          inner else block statements;
       }
  }
else
  {
     outer else block statements;
  }
After nested if block statements;
```

④ Flowchart:



Condition 1
N
outer else block statements;
Y
Condition 2
Y
inner if block statement;
N
inner else block statements;
After if block statements;

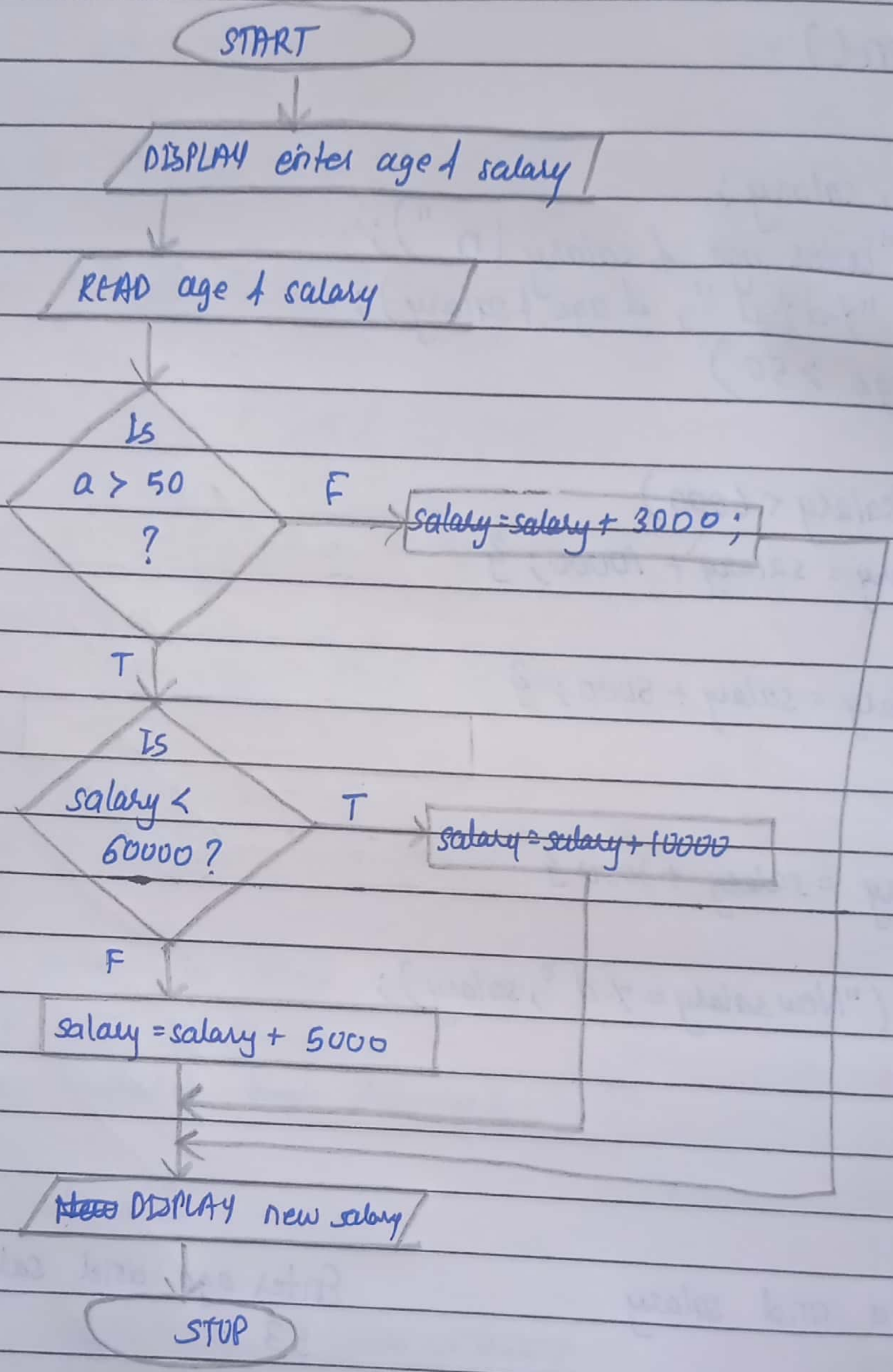
Date.   No.

Eg:
```
# include <stdio.h>
void main()
{
 int age, salary;
 printf ("Enter age & salary \n ");
 scanf ( "%d %d ", & age, & salary);
 if  (age >50)
 {
   if  (salary < 6000)
   { salary = salary + 10000; }
  else
   { salary = salary + 5000 ; }
 }
else
  { salary  = salary  + 3000 }
}
 printf ("New salary = %d ", salary);
 }
```

⊛ Output:

```
Enter age and salary                 Enter age and salary
53                                    53
45000                                 68000
New salary = 55000                    New salary = 73000
```

(*) Flowchart:



Note: To avoid use of nested if, we can use logical operators as necessary.

Date. No.

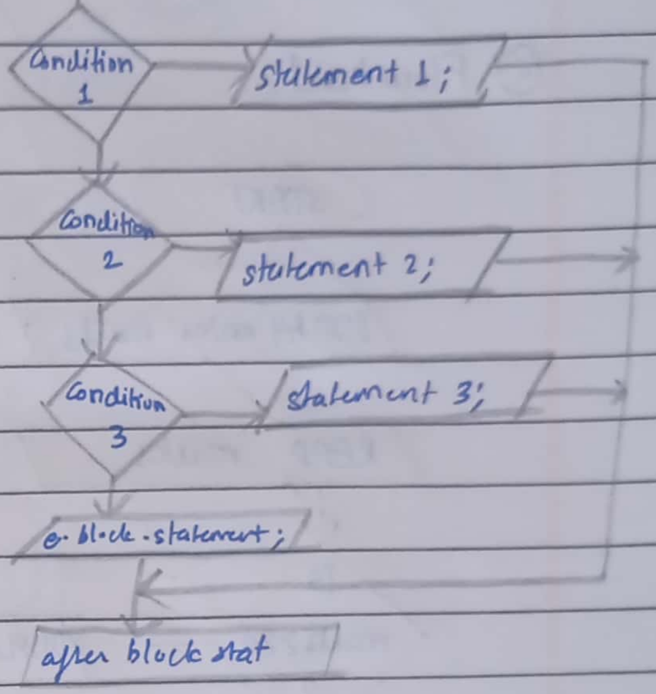
(iv) else-if ladder:

→ It is multipath decision making statement

*) Syntax:

```
if (condition 1)
    { statement 1; }
else if (condition 2)
    { statement 2; }
 else if (condition 3)
   { statement 3; }
else
{ else block statements; }
}
after block statements;
```

*) Flowchart:



- This checks conditions from top to bottom.
→ We can use nested if to avoiding use of else-if ladder.

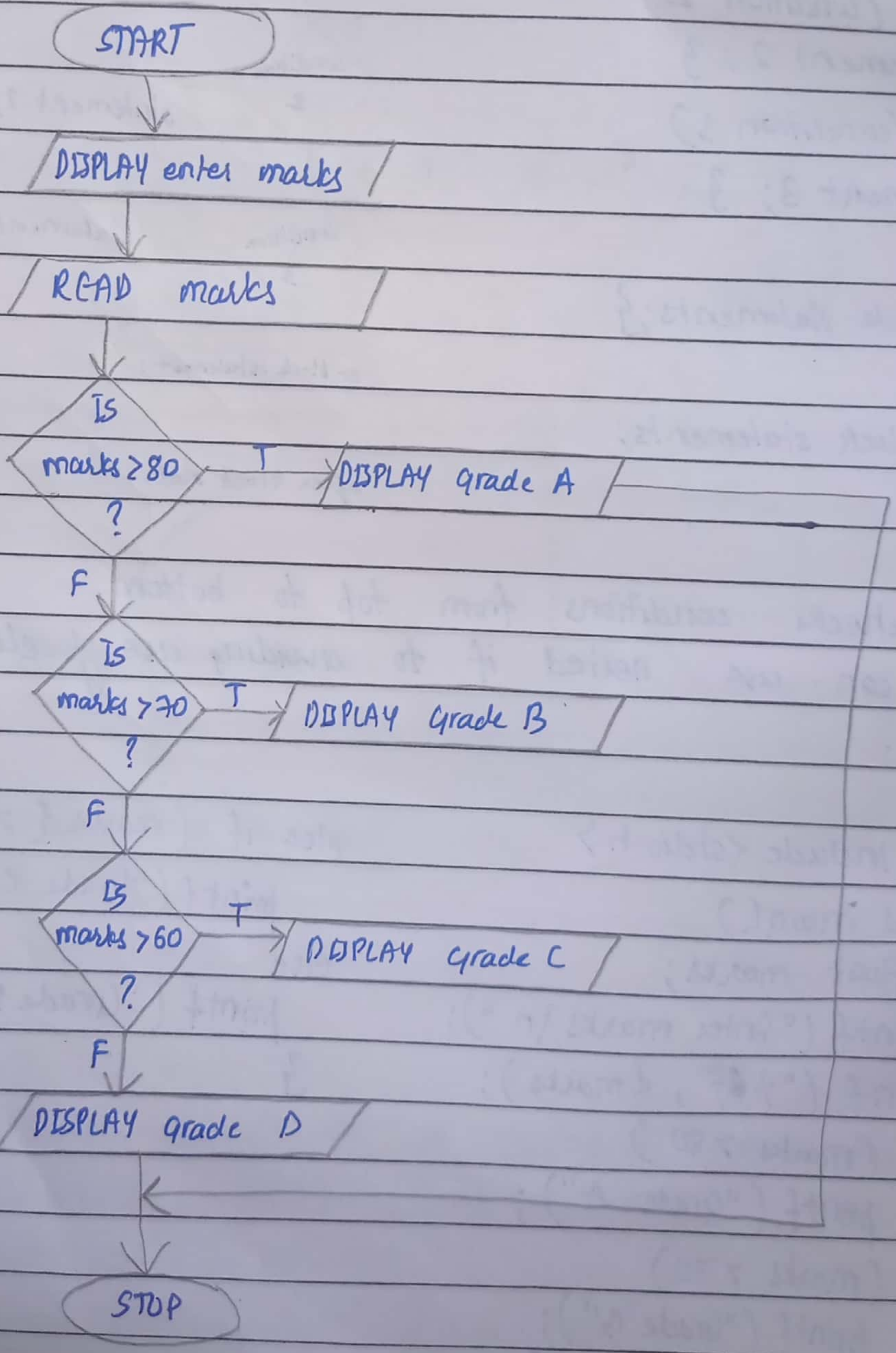
Eg: i:

```
# include <stdio.h>
    void main()
    { float marks;
      printf ("Enter marks \n");
      scanf ("%f", &marks);
     if (marks >80 )
          printf ("Grade A");
  else if (marks >70)
          printf ("Grade B");
else if (marks >60)
     printf ("Grade C");
else
     printf ("Grade D");
}
```

⊛ Output:

```
Enter marks                         Enter marks
89.68                               68.01
A                                   C
```

⊛ Flowchart:



Date.   No.

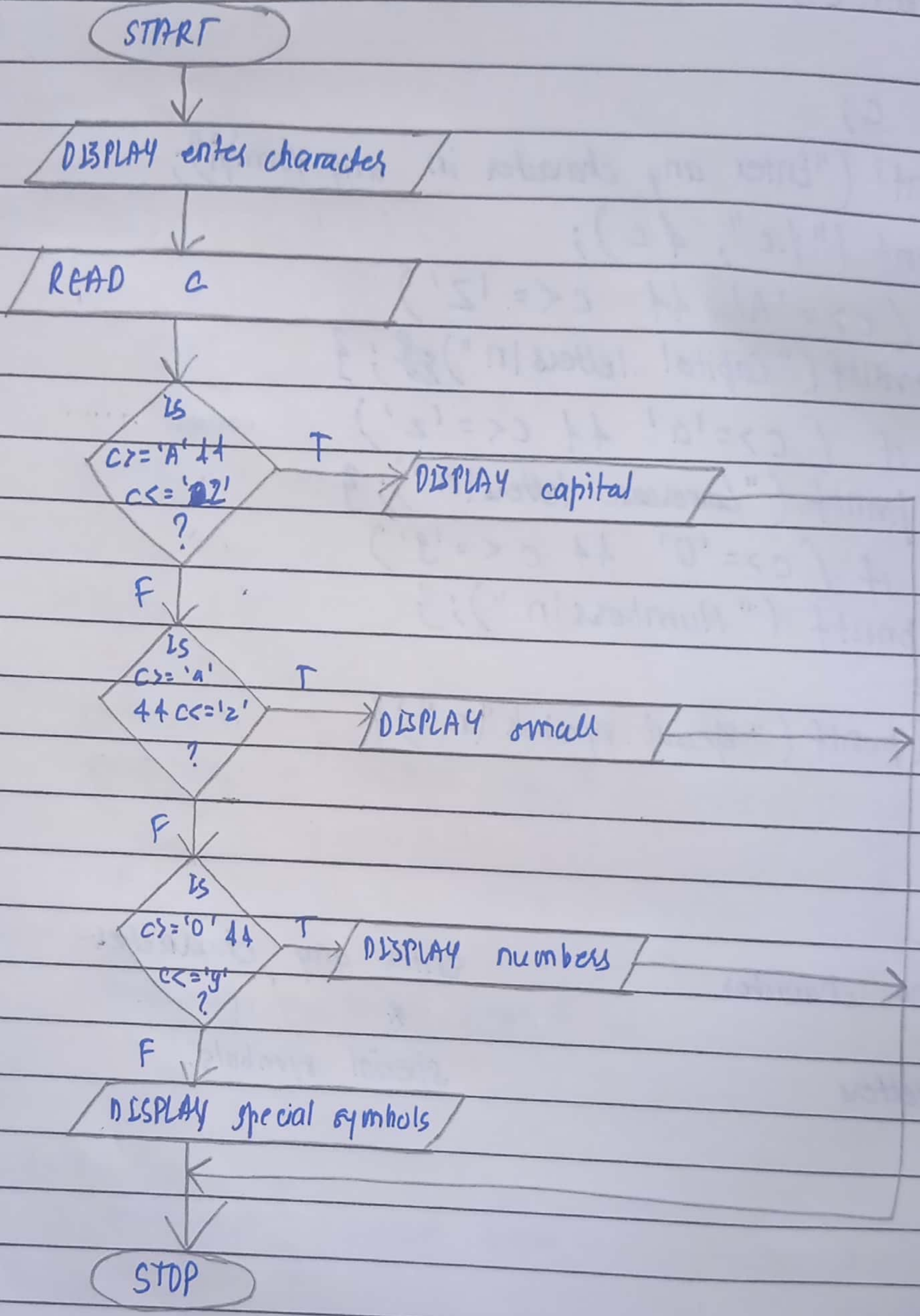
(ii):
```
#include<stdio.h>
 void main()
 {
  char c;
  ~~pr~~ printf("Enter any character in any form\n");
   scanf("%c", &c);
   if (c>='A' && c<='Z')
    {printf("Capital letters\n")~~;~~; }
  else if (c>='a' && c<='z')
   { printf("Lowercase letters\n"); }
  else if (c>='0' && c<='9')
   { printf ("Numbers\n"); }
   else
    {printf("special symbols\n")}
  }
```

⊛ Output:

```
Enter any character              Enter any character
 B                                     *
Capital letters                  special symbols.
```

Date. No.

⊛ Flowchart:



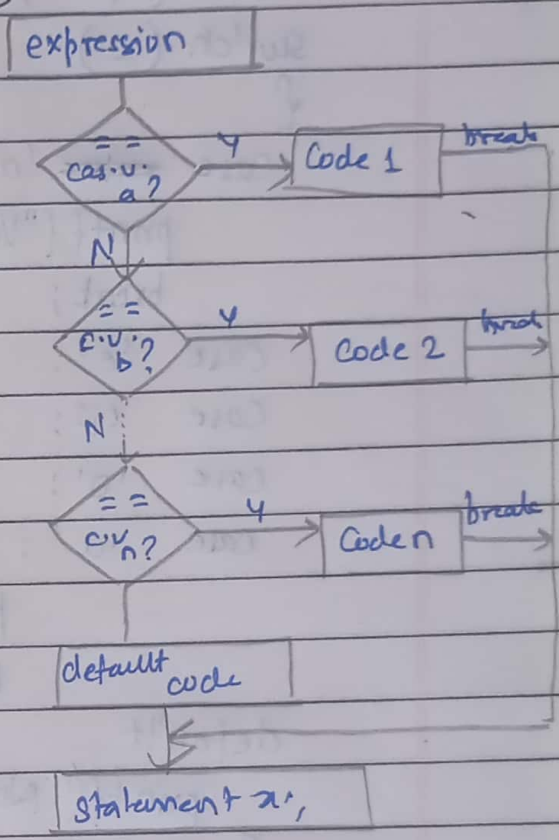
(v) switch

→ It is multi-way ~~deci~~ decision taking statement.
→ It replaces the long if-else version and executes faster than if-else ladder.

(*) Syntax:

```
switch (expression)
  {
     case value a:
          block statement a;
          break;
     case value b:
          block statement b;
          break;
     :
     default
          default statement;
  }
statement x;
```

(*) Flowchart:



Here, case value takes only character or integer value
break: takes you out of the switch.

Not writing break means all statements will be executed.

→ default can be written anywhere and it is not complusory.
→ default is executed at last. → we can't check with relational and logical operators.

Date. No.

Eg:
```
#include <stdio.h>
void main()
{
 char c;
 printf("Enter a character\n");
 scanf("%c", &c);
 switch (c)
 {
  case ~~value~~ 'a' :
      printf("Vowel");
      break;
  case 'e':
  case 'i':
  case 'o':
  case 'u' :
          printf("Vowel")
          break;
 default
     printf("Not a vowel");
 }
```

⇒ we can use <string.h> header file to convert to upper/lower case and check.
+ more suitable.

⊛ Output:

```
Enter a character:
 c
Not a vowel
```

```
Enter character
 a
Vowel.
```

पाठशाला