BLDEA'S

VACHANA PITAMAHA DR.P.G.HALAKATTI COLLEGE OF ENGINEERING AND TECHNOLOGY

DEPT.OF COMPUTER SCIENCE AND ENGINEERING

TITLE: PROBLEM SOLVING THROUGH PROGRAMMING

COURSE COORDINATOR:

PROF.GAYATRI.B

REPORT BY:

1.TEJASHWINI PATIL

2.WAZEERA KHAN

3.VEENA ISAREDDY

4.VEENA MATH

5.Sushma Biradar

INDEX:

1.Using switch statement program to display menus of ATM.

Abstract :

The ATM Program in C is written in C programming language which provides an ease to read and comprehend the instructions used. This program for using ATM machine is built on the concept of handling an account individually.

From this ATM program in C, we can even use the mini-program for checking the total balance, depositing the amount, and withdrawing the amount from the account definitely since it is not time overwhelming.

What is switch statement….??????

*A **switch** statement allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being switched on is checked for each **switch case**.*

The following rules apply to a **switch** statement −

- *The **expression** used in a **switch** statement must have an integral or enumerated type, or be of a class type in which the class has a single conversion function to an integral or enumerated type.*
- *You can have any number of case statements within a switch. Each case is followed by the value to be compared to and a colon.*
- *The **constant-expression** for a case must be the same data type as the variable in the switch, and it must be a constant or a literal.*
- *When the variable being switched on is equal to a case, the statements following that case will execute until a **break** statement is reached.*
- *When a **break** statement is reached, the switch terminates, and the flow of control jumps to the next line following the switch statement.*
- *Not every case needs to contain a **break**. If no **break** appears, the flow of control will fall through to subsequent cases until a break is reached.*
- *A **switch** statement can have an optional **default** case, which must appear at the end of the switch. The default case can be used for performing a task when none of the cases is true. No **break** is needed in the default case.*

Syntax :

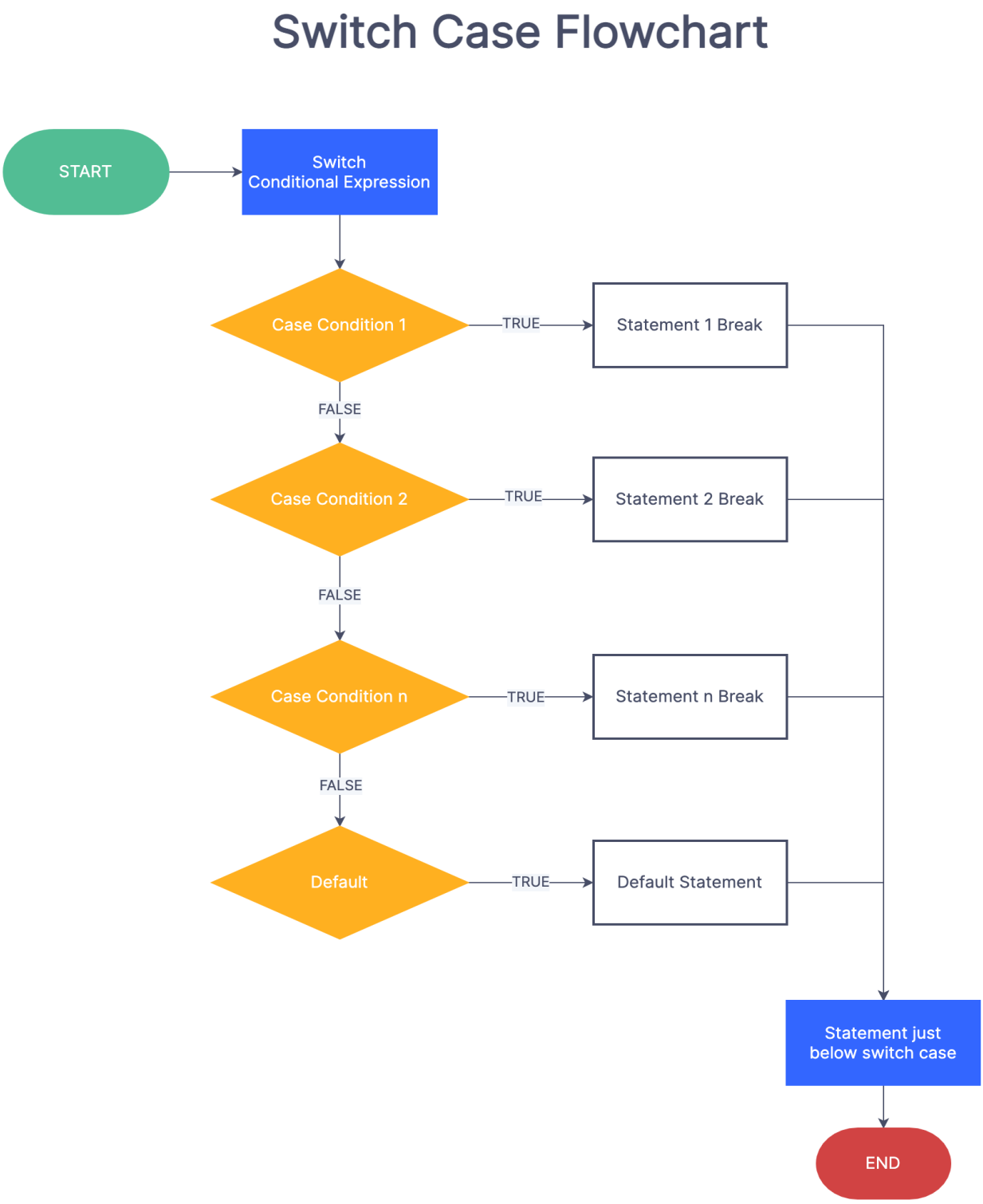
```
switch(expression)
{
case value1:    statement_1; break;

case value2:    statement_2; break;

.....
......
......
case value_n:    statement_n; break;


default:     default statement;

}
```

Some important keywords:

1) Break: *This keyword is used to stop the execution inside a switch block. It helps to terminate the switch block and break out of it.*

2) Default: *This keyword is used to specify the set of statements to execute if there is no case match.*

Note: *Sometimes when* ***default*** *is not placed at the end of switch case program, we should use* ***break statement*** *with the default case.*



*A **switch case flowchart** describes program execution via a graphical representation for simplifying computer programming languages. By displaying a consistent logical sequence between code blocks, the chart brings an easy way to manage multiple cases. This is one of the use cases of flowchart in programming.*

Algorithm

Step1:start

Step2:Read x,y;

Step3:Calculate the answer depending on the operatorand print it

Enter the c for credit,b for balance,d for debit

Enter the credit amount

If x=x+y

Print answer=net amount and go to step4

Enter debit amount

Ifx>=y print ans=net amount and go to step4

If x=x-y

Print answer=net amount

Else

Print ans=insufficient amount go to step4

Else print choose correct option for operation

Goto tryagain

Step4:stop

Program:

```
#include<stdio.h>
int main()
{
Float x,y;
Char ch;
Printf("Enter \n c for credit \n d for debit\n b for balance\n");
Scanf("\n%c",&ch);
Switch(ch)
{
Case 'c':
Printf("Enter the credit amount:\n");
Scanf("\%f",&y);
x=x+y
printf("net amount=%f",x);
break;
case 'd':
printf("Enter debit amount:\n");
scanf("%f",&y);
```

```
if(x>=y)
{
x=x-y
printf("net amount=%f",x);
}
else
{
Printf("insufficient amount=%f",x);
}
break;
case 'b':
printf("amount in account:%f",x);
break;
default:
printf("choose correct option for operation");
goto tryagain;
}
Return(0);
}
```

Output:

b Enter initial amount

5000

Enter

c for credit

d for debit

b for balance

c

Enter credit amount

2000

New amount=7000.00

Enter initial amount

5000

Enter

c for credit

d for debit

b for balance

d

Enter debit amount

3000

New amount=2000.00

Enter initial amount

6000

Enter

c for credit

d for debit

b for balance

d

Enter debit amount

8000

Insufficient balance in your account

Enter initial amount

4500

Enter

c for credit

d for debit

b for balance

Amount in your account=4500.

2.Program to find vowels using switch statement

Abstract:

To check whether the input alphabet is a vowel or consonant in C programming, you have to ask from user to enter a character and check if the given character is equal to a, A, e, E, i, I, o, O, u, U or not. If it is equal to any one of these 10, then it is a vowel, otherwise it is a consonant. Let's have a look at the program:

Algorithm:

Step 1.: Start

Step 2.: Declare character type variable ch.

Step 3.: Read ch from user.

Step 4.: Checking vowels.

If (ch==a

Ch==e

Ch==i

Ch==0

Ch==u)

Print"vowel"

Else

Print"consonant"

Step 5.: stop

Program:

```
#include<stdio.h>
Void main()
{
char alphabet;
printf("Enter an alphabet\n");
scanf("%c",&alphabet);
switch(alphabet)
{
Case 'a':
Printf("It is vowel");
break;
Case 'e':
Printf("It is vowel");
break;
Case 'i':
```

```
Printf("It is vowel");
break;

Case 'o':
Printf("It is vowel");
break;
Case 'u':
Printf("It is vowel");
break;
default:
printf("It is a consonant");
}
getch();
}
```

Output:

Enter an alphabet

a

It is vowel

Enter an alphabet

e

It is vowel

Enter an alphabet

i

It is vowel

Enter an alphabet

o

It is vowel

Enter an alphabet

u

It is vowel

Enter an alphabet

b

It is consonant

Conclusion:

A switch statement allows a variable to be tested foe equality against a list of value.Each value is called a case, and the variable being switched on is checked for each switch case. C is most useful foe embedded systems, or applications that require the ability to be light-weight and have precise control over system resources.