Programing Methodology in C

Lecture 7 – More on Decision Making Using Loops, and Sample Programing Project

By Elubu Joseph

MSc.IS

josebulinga@gmail.com

Agenda

1. Recap on Decision Making using
    i. While loop() and then continue with do while and for loop functions
2. Go through Mobile Application Transaction practical Project.

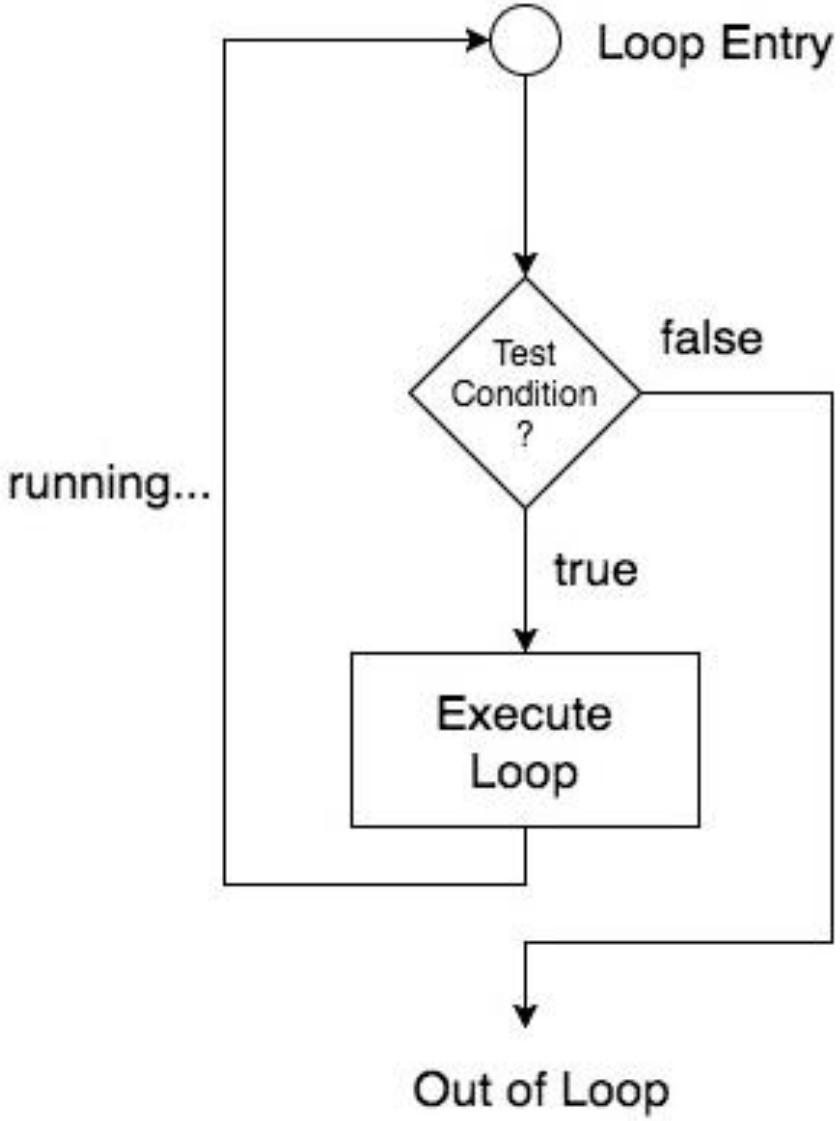
Decision making using Loops in C

How to use Loops in C

In any programming language including C, loops are used to execute a set of statements repeatedly until a particular condition is satisfied.

How it Works

The diagram below depicts a loop execution,



How Loops work explained

As per the above diagram, if the Test Condition is true, then the loop is executed, and if it is false then the execution breaks out of the loop. After the loop is successfully executed, the execution again starts from the Loop entry and again checks for the Test condition, and this keeps on repeating.

The sequence of statements to be executed is kept inside the curly braces { } known as the **Loop body**. After every execution of the loop body, **condition** is verified, and if it is found to be **true** the loop body is executed again. When the condition check returns **false**, the loop body is not executed, and execution breaks out of the loop.

Types of Loop

There are 3 types of Loops in C language, namely:

1. while loop
2. do while loop
3. for loop

Basic Structure of while loop

While loop can be addressed as an **entry control** loop. It is completed in 4 steps.

1. Variable initialization.(e.g int x = 0;)
2. condition(e.g while(x <= 10))
3. Statement to be executed
4. Variable increment or decrement

   (x++ or x-- or x = x + 2)

Syntax :

```
int x=4;
while(x<=10) {
printf("I love it.\n");
x++;
}
```

Output: I love it . 7 times

Output of the above Code

```
I love it.
I love it.
I love it.
I love it.
I love it.
I love it.
I love it.
```

while loop Example: Program to print first 10 natural numbers

```
#include<stdio.h>
void main( ) {
     int x = 1;
     while(x <= 10) {
          printf("%d\t", x);
/* below statement means, do x = x+1,
increment x by 1*/
          x++;
     }
}
```

Output

1 2 3 4 5 6 7 8 9 10

Jumping Out of Loops

Sometimes, while executing a loop, it becomes necessary to skip a part of the loop or to leave the loop as soon as certain condition becomes **true**. This is known as jumping out of loop. We use two word in this process, that is: -

1) break statement

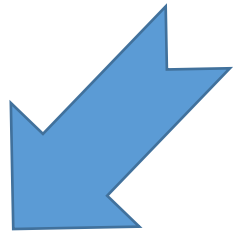
2) Continue statement

1) Break statement

When break statement is encountered inside a loop, the loop is immediately exited and the program continues with the statement immediately following the loop.

```
while( condition check )
{
   statement-1;
   statement-2;
   if( some condition)
   {
      break;
   }
   statement-3;
   statement-4;
}
```

Jumps out of the loop, no matter how many cycles are left, loop is exited.



Sample Break code

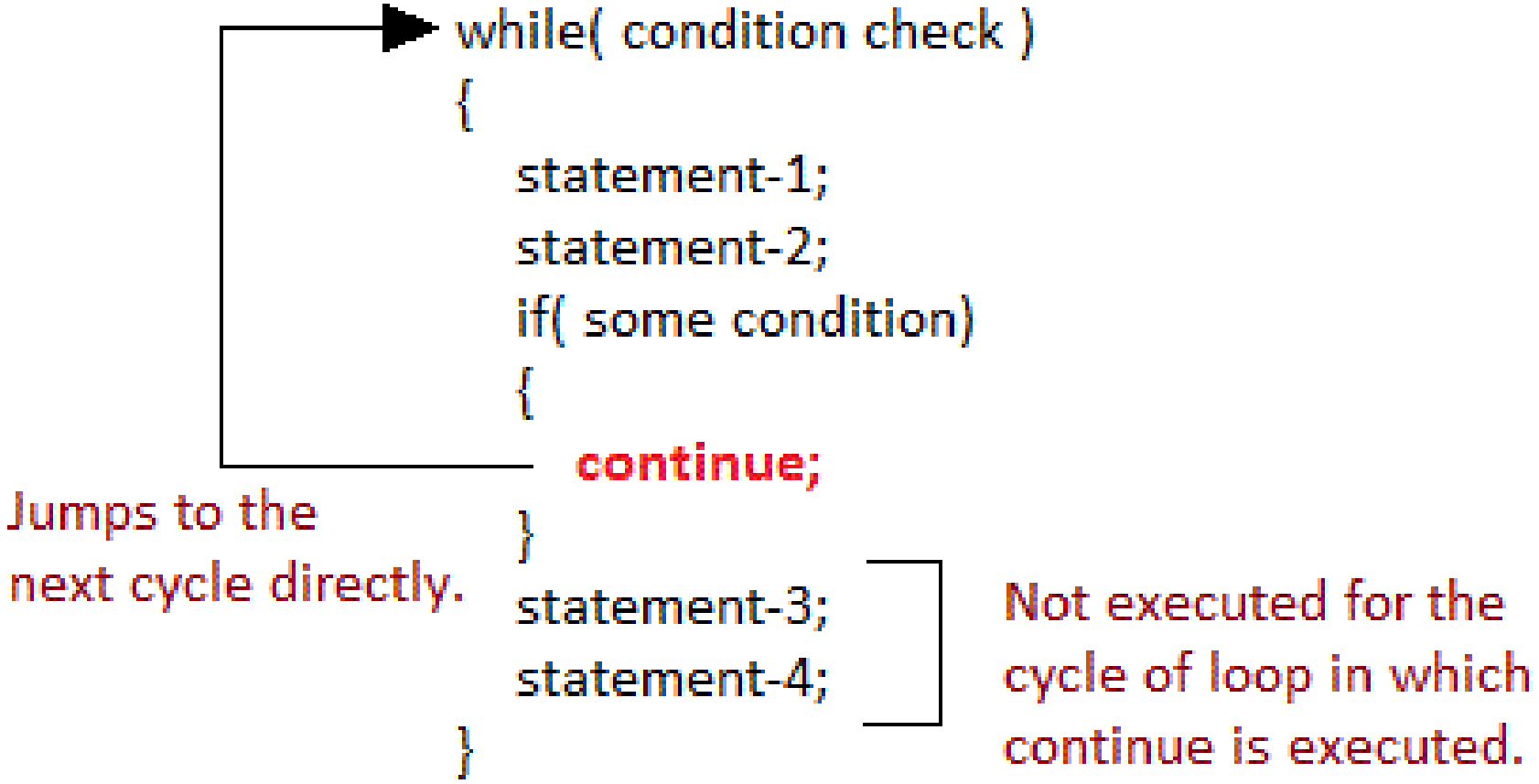
```
#include<stdio.h>
void main( ) {
        int x; x = 1;
        while(x <=10){
                printf("%d\t", x);
        x++;
        if(x<=5){
        break;
        }
         printf("\nNow at ");
        }
        printf("\nNow out of the loop ");
}
```

OUTPUT

```
1
Now out of the loop
```

2) continue statement

It causes the control to go directly to the test-condition and then continue the loop process. On encountering continue, cursor leave the current cycle of loop, and starts with the next cycle.



Sample Continue code

```
#include<stdio.h>
void main( ) {
        int x; x = 1;
        while(x <=10){
                printf("%d\t", x);
        x++;
        if(x<=5){
        continue;
        //break;
        }
          printf("\nNow at ");
        }
        printf("\nNow out of the loop ");
}
```

OUTPUT

```
1       2       3       4       5
Now at 6
Now at 7
Now at 8
Now at 9
Now at 10
Now at
Now out of the loop
```

do while loop Structure Vs while loop

do statement evaluates the body of the loop first and at the end, the condition is checked using while statement. It means that the body of the loop will be executed at least once, even though the starting condition inside while is initialized to be **false**.

General syntax is,

```
do {
Statement block;
Increment statement;
}
while(condition);
```

Vs

```
while(condition){
Statement block;
Increment statement;
}
```

Basic Structure of do while loop

Do while loop is also completed in 4 steps.

1. Variable initialization.(e.g int x = 0;)
2. Statement to be executed
3. Variable increment or decrement
   (x++ or x-- or x = x + 2)
4. condition(e.g while(x <= 10))

Syntax :

```
int x=4;
do{
printf("I love it\n");
x++;
}
while(x<=10);
```

Output: I love it * 7

do loop

Example: Program to print first 10 natural numbers

```
#include<stdio.h>
void main( ) {
     int x; x = 1;
     do{
          printf("%d\t", x);
/* below statement means, do x = x+1,
increment x by 1*/
          x++;
     }
      while(x <= 10);
}
```

Output

1 2 3 4 5 6 7 8 9 10

Example: Program to print first 10 multiples of 5.

```c
#include<stdio.h>
void main() {
    int a, i; a = 5; i = 1;
    do {
        printf("%d\t", a*i);
        i++;
    }
    while(i <= 10);
}
```

OUTPUT:

5 10 15 20 25 30 35 40 45 50

for Loop

for loop Structure

for loop is used to execute a set of statements repeatedly until a particular condition is satisfied. for loop is an **open ended loop.**. General format is,

```
for(initialization; condition; increment/decrement) {

statement-block;

}
```

for loop Structure +

In for loop we have exactly two semicolons, one after initialization and second after the condition.

In this loop we can have more than one initialization or increment/decrement, separated using comma operator. But it can have only one **condition**.

The for loop is executed as follows:

1. It first evaluates the initialization code.
2. Then it checks the condition expression.
3. If it is **true**, it executes the for-loop body.
4. Then it evaluate the increment/decrement condition and again follows from step 2.
5. When the condition expression becomes **false**, it exits the loop.

Example: Program to print first 10 natural numbers

```c
#include<stdio.h>
void main( ) {
int x;
for(x = 1; x <= 10; x++) {
    printf("%d\t", x);
}
}
```

Output

1 2 3 4 5 6 7 8 9 10

Nested for loop

We can also have nested for loops, i.e one for loop inside another for loop. Basic syntax is,

```
for(initialization; condition; increment/decrement)
{
      for(initialization; condition; increment/decrement)
        {
       statement ;
      }
}
```

Example: Program to print half Pyramid of numbers

```
#include<stdio.h>
void main( ) {
     int i, j;
     /* first for loop */
     for(i = 1; i < 5; i++) {
          printf("\n");
     /* second for loop inside the first */
     for(j = i; j > 0; j--) {
               printf("%d", j);
          }
     }
}
```

OUTPUT:

1
21
321
4321
54321

Electronic Mobile Transaction Application programing Project

Electronic Mobile Transaction Application programing Project

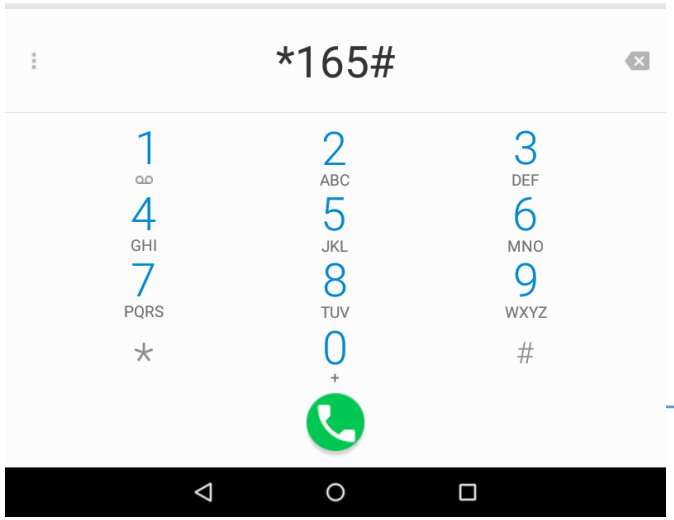
In a number of countries including Uganda, Mobile Electronic transactions is currently transforming the way people do transactions using their phones.

Mobile phones users will need to buy airtime, make various forms of payments, transfer money from their bank accounts to the mobile handsets etc.

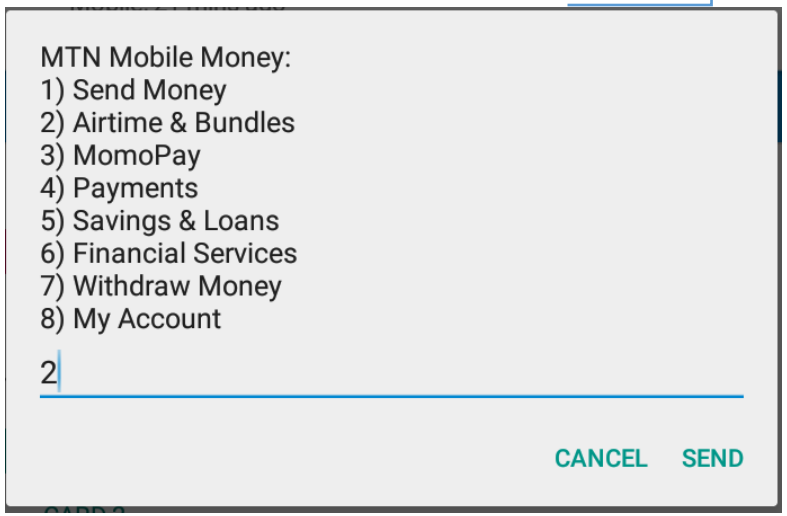
To put this into perspective, we will simulate MTN's Mobile application transaction process as seen below.

MTN Mobile Money Transaction Menu/steps

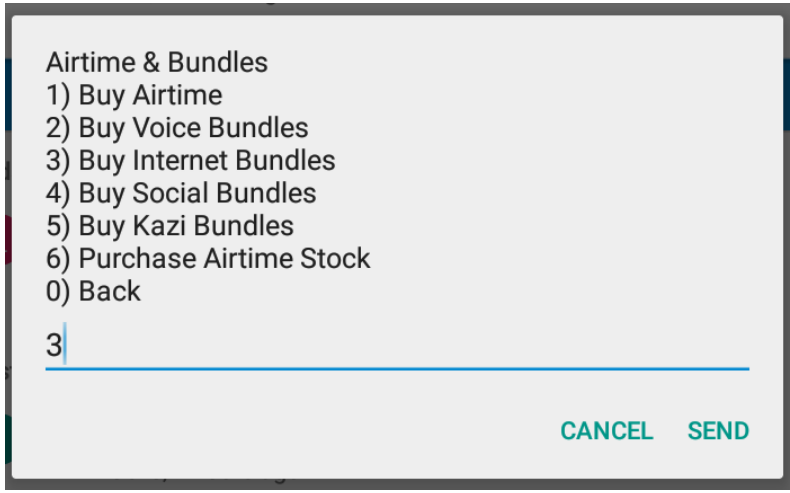
Lets think of a system that a allow a mobile user to buy electronic airtime/voice bundles. The user will have to start from some where and that is e.g. *165# for MTN Uganda & *185# for Airtel Uganda.



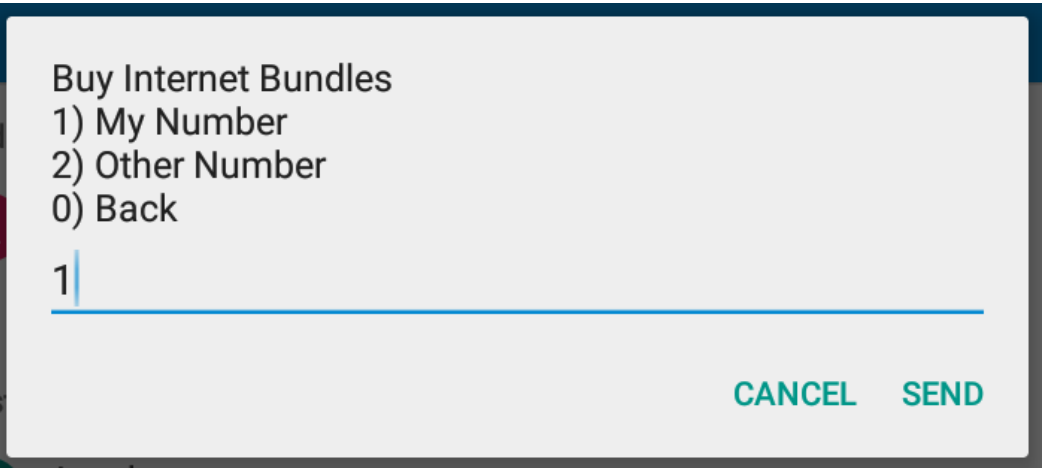
1



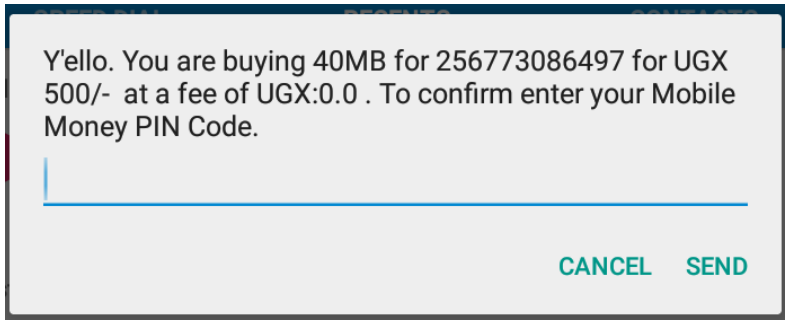
2



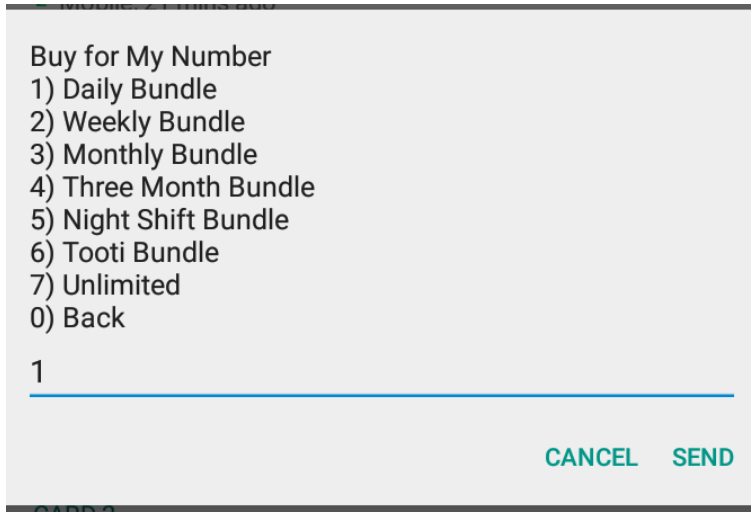
3


Buy Internet Bundles
1) My Number
2) Other Number
0) Back
1
CANCEL
SEND

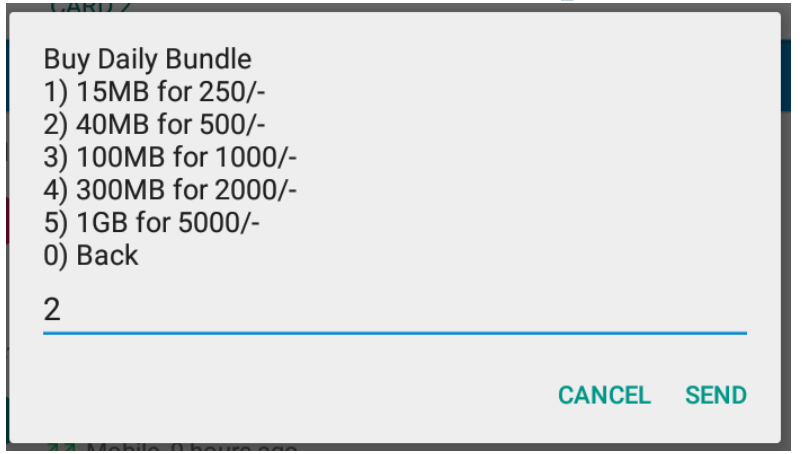
4


Y'ello. You are buying 40MB for 256773086497 for UGX 500/- at a fee of UGX:0.0 . To confirm enter your Mobile Money PIN Code.
CANCEL
SEND

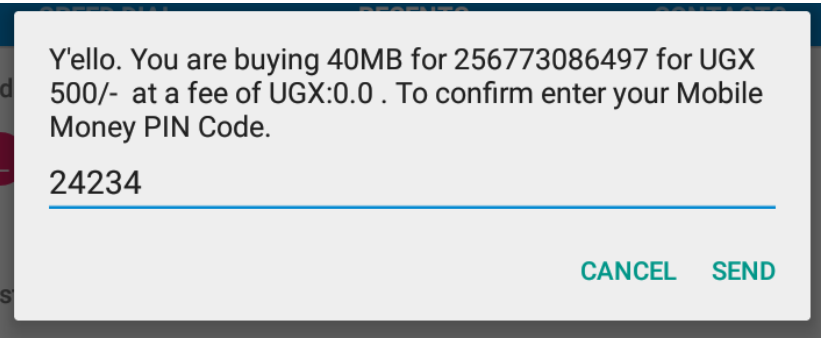
7


Buy for My Number
1) Daily Bundle
2) Weekly Bundle
3) Monthly Bundle
4) Three Month Bundle
5) Night Shift Bundle
6) Tooti Bundle
7) Unlimited
0) Back
1
CANCEL
SEND

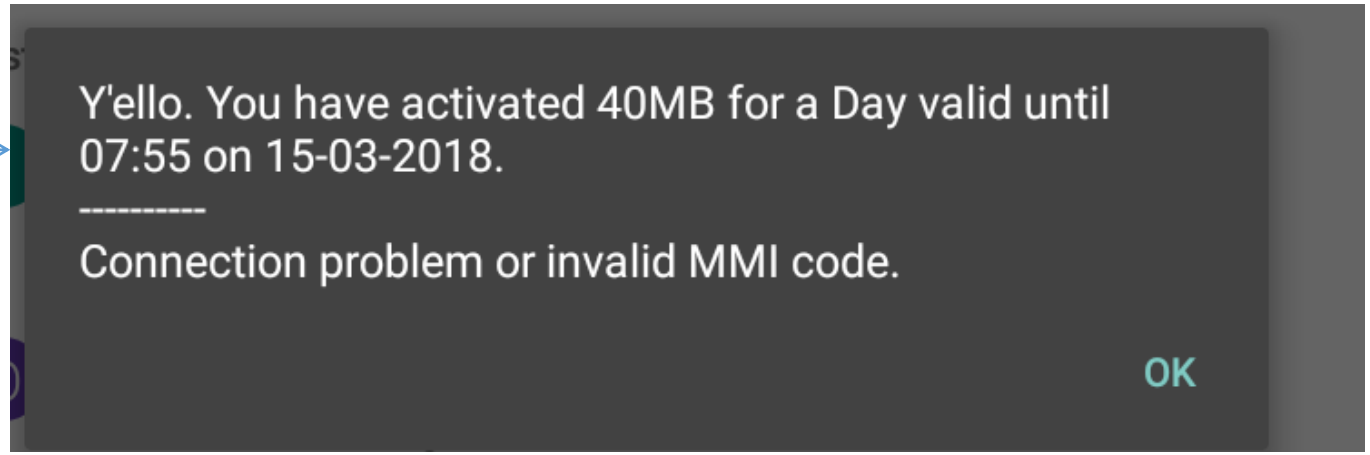
5


Buy Daily Bundle
1) 15MB for 250/-
2) 40MB for 500/-
3) 100MB for 1000/-
4) 300MB for 2000/-
5) 1GB for 5000/-
0) Back
2
CANCEL
SEND

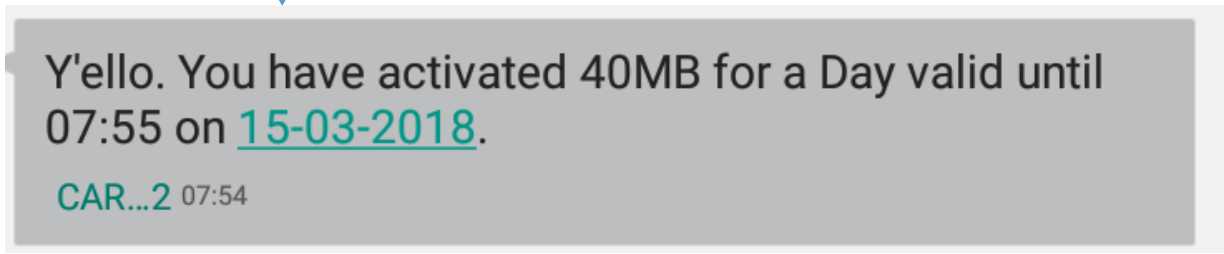
6



8



9 Final Stage



Lets Code now…….

```
#include<stdio.h>
      int answer, num2;
      char start[5];
      int phonenumber=773086497;
      int option; int mynum;

void main( ) {
     printf("\nTo Access MTN Service Dial *165# &
press Enter Key\n\n ");
scanf("%s",start);
```

Lets Code now…….+

```
if(strcmp(start,"*165#")==0){
    printf("\nWelcome to MTN Mobile Money\n------------\n");
    printf("1) Send Money.\n 2) Airtime & Bundles\n 3)
MomoPay\n 4) Payments\n5) Savings & Loans\n6) Financial
Services\n7) Withdraw Money\n 8) My Account\n");
    scanf("%d", &answer);
    if(answer==1){
        printf("Send Money\n--------------------------
\n");

    }
```

Lets Code now…….++

```
else if(answer==2){
                        buyAirtime_Bundles();

            }else if(answer==3){
            printf("MomoPay\n-------------------------\n");
            //printf("1) Buy Bundles\n 2) Buy Voice
Bundles\n 3) Buy Internet Bundles\n 4) Buy Social Bundles\n5) Buy
Kazi Bundles\n6) Buy Airtime Stock\n0) \n");
            scanf("%d", &answer);
            }
            else if(answer==4){
```

Lets Code now…….

```
printf("Payments\n-------------------------\n");
            //printf("1) Buy Bundles\n 2) Buy Voice Bundles\n 3) Buy
Internet Bundles\n 4) Buy Social Bundles\n5) Buy Kazi Bundles\n6) Buy
Airtime Stock\n0) \n");
            scanf("%d", &answer);
            }
            else if(answer==5){
           // printf("Savings & Loans\n----------------\n");
            //printf("1) Buy Bundles\n 2) Buy Voice Bundles\n 3) Buy
Internet Bundles\n 4) Buy Social Bundles\n5) Buy Kazi Bundles\n6) Buy
Airtime Stock\n0) \n");
            scanf("%d", &answer);
            }
```

Lets Code now…….

```
printf("Payments\n----------------------------------------\n");
            //printf("1) Buy Bundles\n 2) Buy Voice Bundles\n 3) Buy Internet Bundles\n 4) Buy Social }
            else if(answer==6){
            printf("Financial Services\n----------------\n");
            //printf("1) Buy Bundles\n 2) Buy Voice Bundles\n 3) Buy Internet Bundles\n 4) Buy Social Bundles\n5) Buy Kazi Bundles\n6) Buy Airtime Stock\n0) \n");
            scanf("%d", &answer);
            }
```

Lets Code now…….

```
else if(answer==7){
            printf("Withdraw Money\n--------------------\n");
             //printf("1) Buy Bundles\n 2) Buy Voice
Bundles\n 3) Buy Internet Bundles\n 4) Buy Social
Bundles\n5) Buy Kazi Bundles\n6) Buy Airtime Stock\n0) \n");
             scanf("%d", &answer);
             }
```

Lets Code now…….

```
else if(answer==8){
            printf("My Account\n--------------------------
------------------");
            //printf("1) Buy Bundles\n 2) Buy Voice
Bundles\n 3) Buy Internet Bundles\n 4) Buy Social
Bundles\n5) Buy Kazi Bundles\n6) Buy Airtime Stock\n");
            scanf("%d", &answer);
            }
```

Lets Code now…….

```
 else{
                printf("Unknown Application");
            }
}
else{
printf(" Not recognized application");
}

getch();
} //end of main function
```

Lets Code now…….

```
int buyAirtime_Bundles(){

printf("Airtime & Bundles\n-------------------------------
--------------\n");
        printf("1) Buy Airtime \n 2) Buy Bundles\n 3) Buy
Voice Bundles\n 4) Buy Internet Bundles\n 5) Buy Social
Bundles\n6) Buy Kazi Bundles\n7) Buy Airtime Stock\n0)
Back\n");
        scanf("%d", &num2);
        if(num2==1){
```

To complete this code, you will need to see the file called MTN_Mobile_Money

This program is quite bulky so see more of this source code in the file mentioned above herein attached.

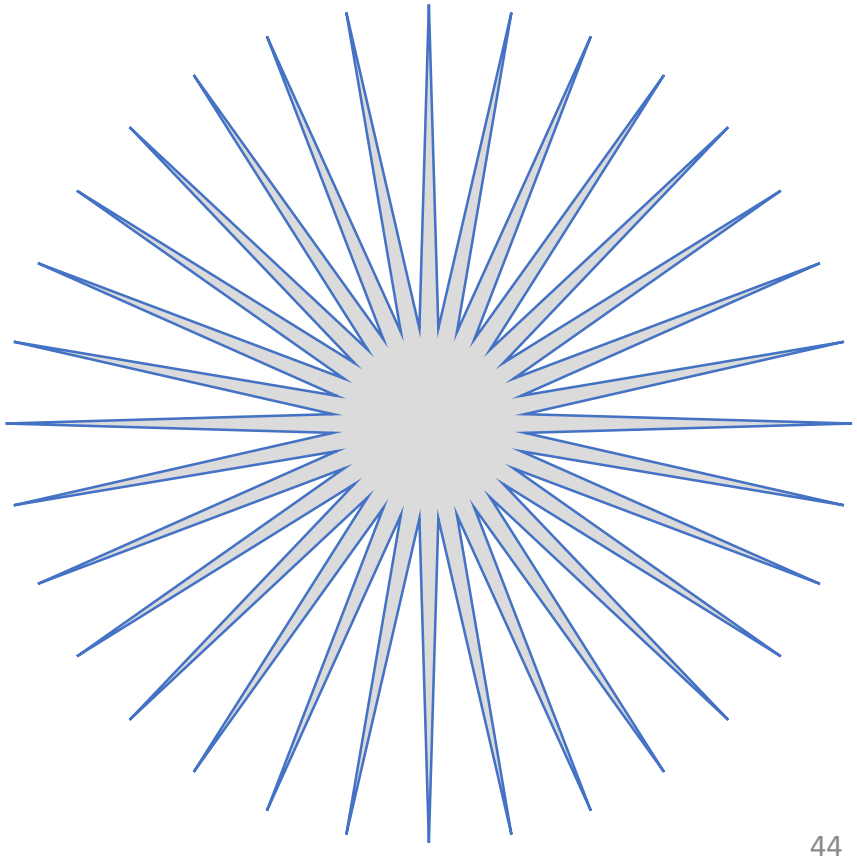
NOTE that we have stop at a user defined function called:

buyAirtime_Bundles(), you may proceed the rest of the codes in the above mentioned source file.

Summary

1. Recap on Decision Making
    i. Loop functions –introduced three loop functions dealt with while loop(layout ant sample program)
    ii. Had Mobile money transaction programing project.

Thank you for you attention



References

Decision making in C. Studytonight.com. (n.d.). Retrieved October 9, 2021, from https://www.studytonight.com/c/decision-making-in-c.php.