Vending Machine

Submitted By:

MUHAMAD TALHA

SP20-BCS-068

Submitted To:

Mr. Taimoor Shehzad

COMSATS UNIVERSITY ISLAMABAD

Contents

INTRODUCTION

A vending machine is a machine that dispenses products, such as snacks or beverages, in exchange for payment. Vending machines are often used in public places to provide a convenient way for people to purchase products without the need for human interaction.

Creating a vending machine in assembly language involves designing and implementing a program that can control the hardware and carry out the various tasks involved in vending products. This includes accepting payment, checking product availability, and dispensing the product. The program will be written in assembly language, a low-level programming language that is used to write programs that are executed directly by the processor.

To successfully complete the project, you will need to have a strong understanding of both the hardware components of the vending machine and the assembly language programming language. You will also need to design algorithms to handle tasks such as calculating change and displaying messages to the user.

GENERAL OVERVIEW:

A vending machine programmed in assembly language is a machine that dispenses products, such as snacks or beverages, in exchange for payment. The machine is controlled by a microprocessor, which executes a program written in assembly language to carry out various tasks such as accepting payment, checking product availability, dispensing the product, and returning change. The vending machine also has various hardware components, such as a display screen, buttons, a coin and bill acceptor, and a product dispensing mechanism, which are controlled by the microprocessor and assembly language program to allow the user to interact with the machine and purchase products.

The assembly language program for the vending machine would include instructions for handling the various tasks involved in vending products, such as accepting payment, checking product availability, and dispensing the product. The program would also include instructions for handling error conditions, such as when the user tries to purchase a product that is out of stock or does not have enough money.

The vending machine would be able to accept payment in the form of cash or credit card and would have the ability to calculate the correct change to return to the user. It would also have a display screen to show messages to the user, such as the price of the selected product or an error message if there is a problem with the transaction.

Overall, the vending machine would provide a convenient and automated way for users to purchase products without the need for human interaction. It would be able to operate 24/7 and would be able to handle multiple transactions simultaneously.

BACGROUND:

Vending machines are typically controlled by a microprocessor, which executes a program to carry out the various tasks involved in vending products. These programs can be written in a variety of programming languages, including assembly language.

Assembly language is a low-level programming language that is used to write programs that are executed directly by the processor. It is considered a "low-level" language because it is closer to the machine language that the processor can execute directly and is more difficult to read and understand than higher-level languages like C or Python.

Assembly language is often used in systems where speed and memory efficiency are important, such as in embedded systems or real-time systems. Vending machines are examples of systems that might benefit from using assembly language because they often have limited resources and require fast response times to handle transactions.

Overall, the use of assembly language in vending machines allows for efficient and reliable control of the machine's hardware and algorithms, which is important for providing a seamless user experience.

DESCRIPTION:

A vending machine programmed in assembly language would be a machine that dispenses products, such as snacks or beverages, in exchange for payment. The machine would accept payment in the form of cash or a credit card and would be able to calculate the correct change to return to the user.

The vending machine would be controlled by a microprocessor, which would execute a program written in assembly language to carry out the various tasks involved in vending products. The program would include instructions for accepting payment, checking product availability, dispensing the product, and returning change.

The vending machine would also have various hardware components, such as a display screen, buttons for selecting products and entering payment, a coin and bill acceptor, and a product dispensing mechanism. These components would be controlled by the microprocessor and the assembly language program to allow the user to interact with the vending machine and purchase products.

DISCUSSION

Visual Studio requires assembly language source files to belong to a project, which is a kind of a box container. A project holds configuration information such as the locations of the assembler, linker, and required libraries. A project has its folder, and it holds the names and locations of all files belonging to it.

This project uses the project template in the form *.asm* given by our lecturer from the beginning of learning assembly language. We are required to use the Irvine Link Library. The Irvine link library contains several useful routines to input data, output data, and perform several tasks that one would normally have to use many operating systems calls to complete. This library is called Irvine. Lib and should have been installed when you installed the CD-ROM from the Irvine book. We only use a few basic things here. Here are list some procedures in the Irvine32 library that we used in making this project:

Clrscr	Clears the screen, moves the cursor to the upper-left corner.
Crlf	Writes a carriage return / linefeed to the display.
Mov	Copies a byte or word from a source operand to a destination operand.
Writestring	Write a null-terminated string. Input: EDX points to the strings offset
Mul	Multiplies AL, AX, or EAX by a source operand.
ReadDec	Reads an unsigned 32-bit decimal integer from the keyboard, terminated by the Enter key.
WriteDec	Writes an unsigned 32-bit integer to the console window in decimal format.
Cmp	Compares the destination to the source by performing an implied subtraction of the source from the destination.
Jmp	*(Jump Unconditionally)*Jump to a code label.
Jz	*(Conditional Jump)* Jump Equal or Jump Zero.
Jge	*(Conditional Jump)* Jump Greater/Equal or Jump Not Less.
Jl	*(Conditional Jump)* Jump Less or Jump Not Greater/Equal.

The loop instruction is frequently used in our project such as L1, L2, FLAV, F1-F4, Cal1-Cal4 and so on. The functions of each loop used in the program are shown in the table

below:

Loop	Function
L1	Display the menu
L2	Input the total number of moneys to be inserted according to the value of money
FLAV	Input an item that choose from the menu
F1-F4	F1 to F4 denoted as the flavours of doughnut as shown in the item banner. Compare the total of money inserted, the instruction will jump to the loop SRY if the money inserted is not enough. Compare the item chosen so that the instruction will jump to the corresponding loop for calculation (Cal1-Cal4).
Cal1-Cal4	Calculate the change after the input of the item chosen.
ASK	Ask the user whether to continue the program or quit.
CONTINUE	Continue the program by restart from the loop L1.
SRY	Display a string showing the money inserted is not enough. The instruction will jump back to L2 and ask the user to input again for the money.
STOP	Stop the program.

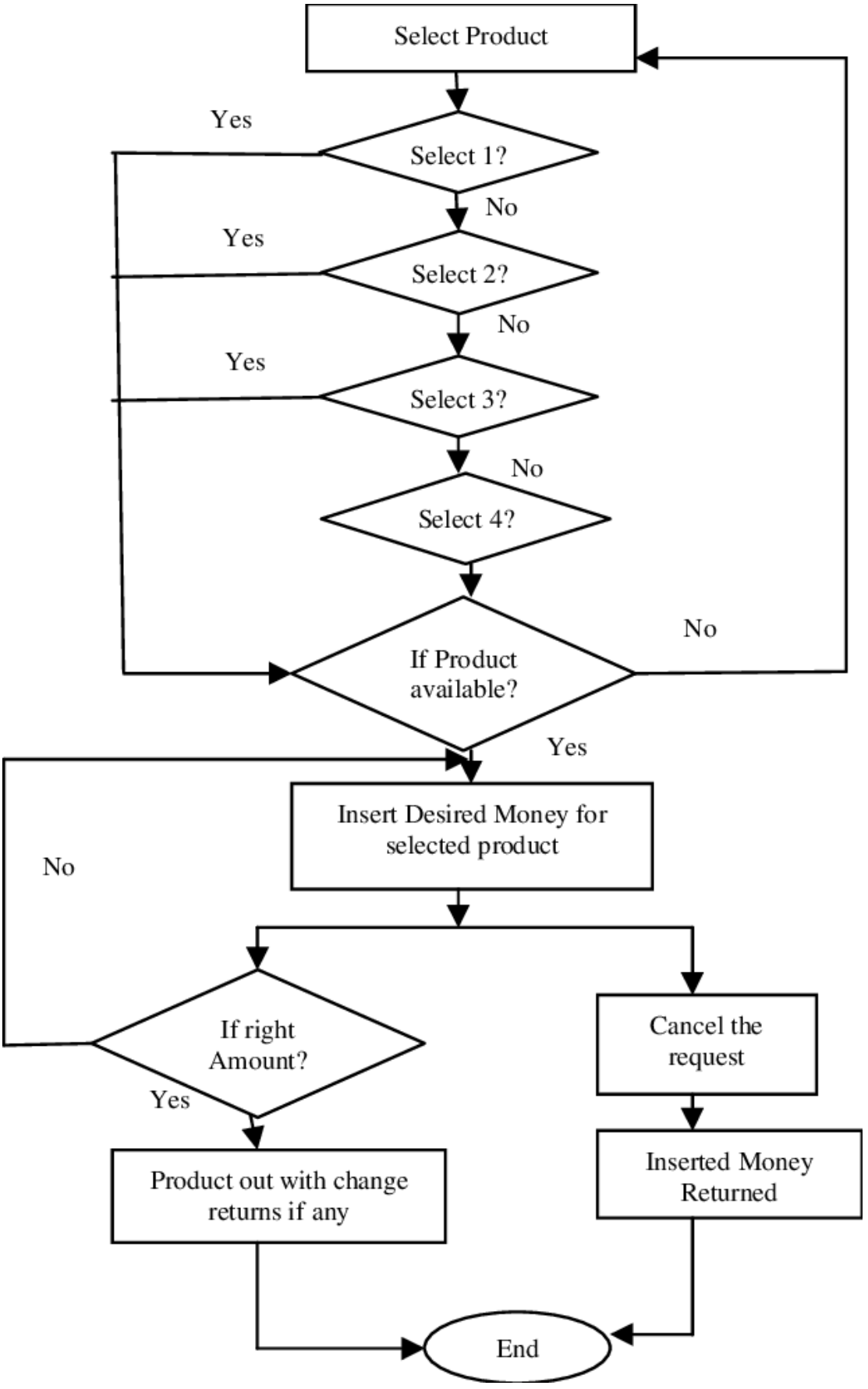
Overall, the goal of the project is to create a vending machine that is reliable, efficient, and user-friendly. This will require careful planning and attention to detail, as well as a strong understanding of assembly language and the hardware components of the vending machine.

In this project, I have tried to apply the knowledge of assembly language that we learnt in the subject computer organization and architecture to complete all the tasks required. The main task of this project is required students to develop a simulation program of a vending machine with different items. We have designed a doughnut vending machine with different flavours and different prices. We assigned prices according to their flavours as shown in the table below:

Item #	Flavour	Price / Rs
1	Original Lays	250
2	Chocolate Cream Cake with Oreo	300
3	Caramel Cream Crunch	200
4	Chocolate Iced Glazed with Sprinkles	200

FLOW CHART:

1. **Initialize**:
   Set up the vending machine by filling it with products, setting the prices, and ensuring that the payment system is working properly.
2. **Wait for user input:**
   The vending machine waits for the user to select by pressing a button corresponding to a particular product.
3. **Check availability**:
   The vending machine checks to see if the selected product is in stock. If it is not, the machine displays an "out of stock" message and returns to the wait state.
4. **Check payment:**
   The vending machine checks to see if the user has entered sufficient payment. If not, it displays a "please insert more money" message and returns to the wait state.
5. **Dispense product:**
   If the product is in stock and the payment is sufficient, the vending machine dispenses the product and returns any remaining change to the user.
6. **Return to wait state:**
   The vending machine returns to the wait state and is ready to process the next transaction.


Select Product
Yes
Select 1?
No
Yes
Select 2?
No
Yes
Select 3?
No
Select 4?
No
If Product available?
Yes
Insert Desired Money for selected product
No
If right Amount?
Yes
Cancel the request
Product out with change returns if any
Inserted Money Returned
End

ADVANTAGE:

There are several advantages to using an assembly language to program a vending machine:

1. **Speed**:
   Assembly language is much faster than high-level languages like C or Python because it is closer to the machine language that the processor can execute directly. This can be important for time-sensitive tasks like vending machine transactions.
2. **Memory efficiency:**
   Assembly language programs typically require less memory than their high-level language counterparts, which can be important for systems with limited memory resources like vending machines.
3. **Control:**
   With assembly language, you have a high level of control over the hardware and can optimize the code for specific features of the processor. This can be important for designing efficient vending machine algorithms.
4. **Debugging:**
   Assembly language programs are easier to debug than machine language programs because they are written in a more human-readable form. This can make it easier to identify and fix problems in the vending machine's code.

WEAKNESS OF THE PROGRAM

The weakness of the coding we made is that it is too long and simple. Much efficient ways or shortcuts for a much simpler coding may be learnt over time through experience. But even so, this code still runs according to the instructions following this project and uses assembly language that we have learned during class sessions.

In addition, one of the problems that we faced while designing the code is that we hope to enable the user to add as many items as possible if the changes are enough for the next items. Example if the user inserted a total of 500 cents. The first item chosen is original glazed flavour doughnut (250 cents). And the change is 250 cents, which should be possible to add another item that is below 250 cents. So, this is the part that we failed to add into our program. And we think that the code would be more perfect if this part is included.

CODE

```
 ;SP20-BCS-068 MUHAMMAD TALHA

;VENDING MACHINE
INCLUDE Irvine32.inc
.data
    D1 BYTE " ------------------------------------------------------------" ,0
    D2 BYTE "| 1. ORIGINAL LAYS 250 Rs                                    |"
,0
    D3 BYTE "| 2. CHOCOLATE CREAM CAKE WITH OREO 300 Rs                   |"
,0
    D4 BYTE "| 3. CARAMEL CREAM CRUNCH 200 Rs                             |"
,0
    D5 BYTE "| 4. CHOCOLATE ICED GLAZED WITH SPRINKLES 200 Rs             |"
,0

    item1 BYTE "Here is your : ORIGINAL GLAZED FLAVOUR DOUGHNUT",0
    item2 BYTE "Here is your : CHOCOLATE CREAM CAKE WITH OREO FLAVOUR
DOUGHNUT",0
    item3 BYTE "Here is your : CARAMEL CREAM CRUNCH FLAVOUR DOUGHNUT",0
    item4 BYTE "Here is your : CHOCOLATE ICED GLAZED WITH SPRINKLES FLAVOUR
DOUGHNUT",0

    str1 BYTE "Please input money : ",0
    str2 BYTE "10 Rs: ",0
    str3 BYTE "20 Rs: ",0
    str4 BYTE "50 Rs: ",0
    str5 BYTE "Total Money Inserted: ",0
    str6 BYTE "Please choose item : ",0
    str7 BYTE "This is your change : ",0
    str8 BYTE "Thank you! ", 0
    str18 BYTE "HELLOOO! ", 0

    welcome BYTE "Hope you enjoyed using our machine! ", 0

    q1 BYTE "Do you want to add item? (Yes:1 / No: (press any key) : ",0
    Sorryy BYTE "YOU DO NOT HAVE ENOUGH MONEY !!! PLEASE TRY AGAIN !",0

    Rs BYTE " Rs",0
    flavour DWORD 0
    input1 DWORD 0
    input2 DWORD 0
    input3 DWORD 0
    total DWORD 0
    prevtotal DWORD 0
    change DWORD 0
    choice DWORD 0
    ten DWORD 10
```

```
    twenty DWORD 20
    fifty DWORD 50
    cost1 DWORD 250
    cost2 DWORD 300
    cost3 DWORD 200
    cost4 DWORD 200

.code
main PROC
call Clrscr
;-------------------------------------------------------------


L1:
    call PrintAll
    jmp L2


;-------------------------------------------------------------
L2:
    call InputAll

    jmp L3

;-------------------------------------------------------------

L3:
    call PrintingTotal



    jmp FLAV


hello:
    mov edx, OFFSET str18
    call WriteString



FLAV:
    mov edx, OFFSET str6
    call WriteString
    call ReadDec
    mov flavour, eax
    cmp flavour,1
    jz Flavour1
    cmp flavour,2
    jz Flavour2
    cmp flavour,3
```

```
    jz Flavour3
    cmp flavour,4
    jz Flavour4
    jmp FLAV

;-------------------------------------------------------------
Flavour1:
    cmp total, 250
    jge Cal1
    jl Sorry
    ;-------------------------------------------------------------


    Cal1:
    mov edx, OFFSET item1
    call WriteString
    call crlf
    mov eax, total
    sub eax, cost1

    mov change, eax
    mov edx, OFFSET str7
    call WriteString
    mov eax, change
    call WriteDec
    mov total, eax

    mov edx, OFFSET Rs
    call WriteString
    call crlf
    mov edx, OFFSET welcome
    call WriteString
    call crlf
    jmp ASK
;-------------------------------------------------------------
Flavour2:
    cmp total, 300
    jge Cal2
    jl Sorry
;-------------------------------------------------------------
Cal2:
    mov edx, OFFSET item2
    call WriteString
    call crlf
    mov eax, total
    sub eax, cost2
    mov change, eax
    mov edx, OFFSET str7
    call WriteString
    mov eax, change
```

```
    call WriteDec

    mov total, eax

    mov edx, OFFSET Rs
    call WriteString
    call crlf
    mov edx, OFFSET welcome
    call WriteString
    call crlf

    ;jmp FLAV

    jmp ASK

;-------------------------------------------------------------
Flavour3:
    cmp total, 200
    jge Cal3
    jl Sorry
;-------------------------------------------------------------

Cal3:
    mov edx, OFFSET item3
    call WriteString
    call crlf
    mov eax, total
    sub eax, cost3
    mov change, eax
    mov edx, OFFSET str7
    call WriteString
    mov eax, change
    call WriteDec
    mov total, eax

    mov edx, OFFSET Rs
    call WriteString
    call crlf
    mov edx, OFFSET welcome
    call WriteString
    call crlf
    ;jmp FLAV
    jmp ASK

;-------------------------------------------------------------
Flavour4:
    cmp total, 200
    jge Cal4
    jl Sorry
```

```
;-------------------------------------------------------
Cal4:
    mov edx, OFFSET item4
    call WriteString
    call crlf
    mov eax, total
    sub eax, cost4
    mov change, eax
    mov edx, OFFSET str7
    call WriteString
    mov eax, change
    call WriteDec
    mov total, eax

    mov edx, OFFSET Rs
    call WriteString
    call crlf
    mov edx, OFFSET welcome
    call WriteString
    call crlf
    ;jmp FLAV
    jmp ASK

;-------------------------------------------------------
ASK:
    call crlf

    mov edx, OFFSET q1
    call WriteString
    call ReadDec
    mov choice, eax
    call crlf
    call crlf
    cmp choice,1
    jz Continue
    jmp Stop
;-------------------------------------------------------
Continue:
    jmp FLAV
    jz Stop
;-------------------------------------------------------
Sorry:
    mov edx, OFFSET Sorryy
    call WriteString
    call crlf
    call crlf
    JMP L2
;-------------------------------------------------------
Stop:
    mov edx, OFFSET str8
```

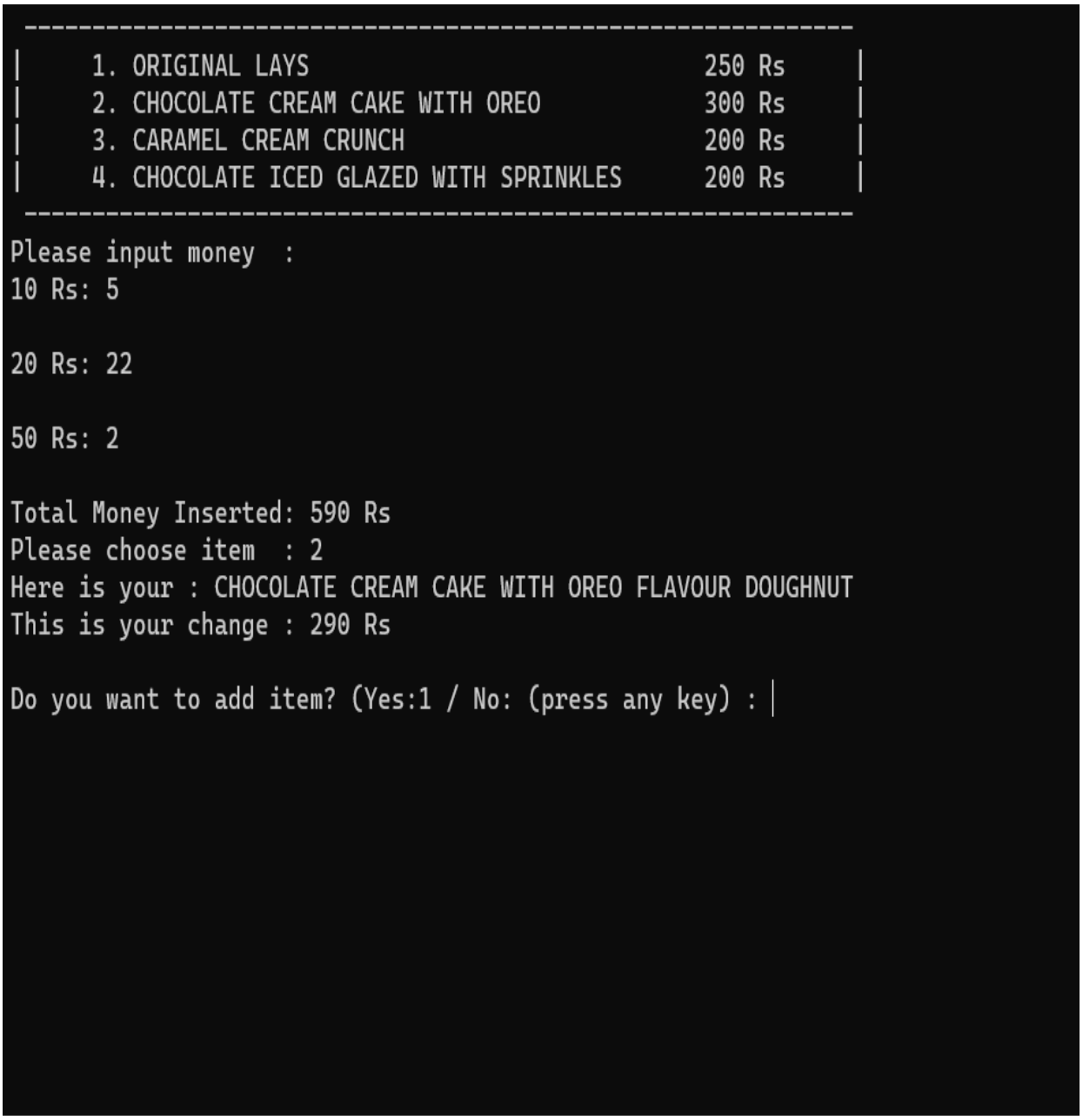
```
    call WriteString
    call crlf
    exit
;-------------------------------------------------------
main ENDP


PrintAll PROC
    mov edx, OFFSET D1   ; ------------------------------------------------
---------

    call WriteString
    call crlf
    mov edx, OFFSET D2  ;| 1. ORIGINAL LAYS 250
Rs                                  |
    call WriteString
    call crlf
    mov edx, OFFSET D3  ;| 2. CHOCOLATE CREAM CAKE WITH OREO 300
Rs                    |
    call WriteString
    call crlf
    mov edx, OFFSET D4  ;| 3. CARAMEL CREAM CRUNCH 200
Rs                              |
    call WriteString
    call crlf
    mov edx, OFFSET D5  ;| 4. CHOCOLATE ICED GLAZED WITH SPRINKLES 200
Rs              |
    call WriteString
    call crlf
    mov edx, OFFSET D1  ;------------------------------------------------
-------
    call WriteString
    call crlf

    ret

PrintAll ENDP


InputAll PROC
    mov edx, OFFSET str1
    call WriteString
    call crlf
    mov edx, OFFSET str2 ;10 Rs:
    call WriteString ; printing it
    call ReadDec
    mov input1, eax
    mov eax, input1  ;initialized variable to check the number of tens stored
in eax
    mul ten   ; Multiplying the number of tens we have to 10 ;10*InputNumber
```

```
    mov total, eax ; as the value of the number of tens is in eax so adding it
into a total variable we have
    call crlf
    mov edx, OFFSET str3  ;20 Rs:
    call WriteString    ; printing it
    call ReadDec
    mov input2, eax   ;initialized variable to check the number of twentys
stored in eax
    mov eax,input2
    mul twenty   ; Multiplying the number of twentys we have to 20
;20*InputNumber
    add total, eax ; as the value of the number of 20s are in eax so adding it
into a total variable we have ; also having the previous tens sum
    call crlf
    mov edx, OFFSET str4  ;50 Rs:
    call WriteString  ; printing it
    call ReadDec ; reading decimal
    mov input3, eax   ;initialized cariable to check the number of fiftys
stored in eax
    mov eax,input3
    mul fifty    ; Multiplying the number of fifty we have to 50
;50*InputNumber
    add total, eax ; as the value of the number of 50s are in eax so adding it
into a total variable we have ; also having the previous twentys sum

    call crlf


    ret

InputAll ENDP




PrintingTotal PROC
    mov edx, OFFSET str5
    call WriteString
    mov eax, total
    call WriteDec
    mov edx, OFFSET Rs
    call WriteString
    call crlf

    ret

PrintingTotal ENDP


END main
```

EXAMPLE OF INPUTS & OUTPUTS

Here is the print screen when we build project and debug:

1. Example of total money inserted is enough



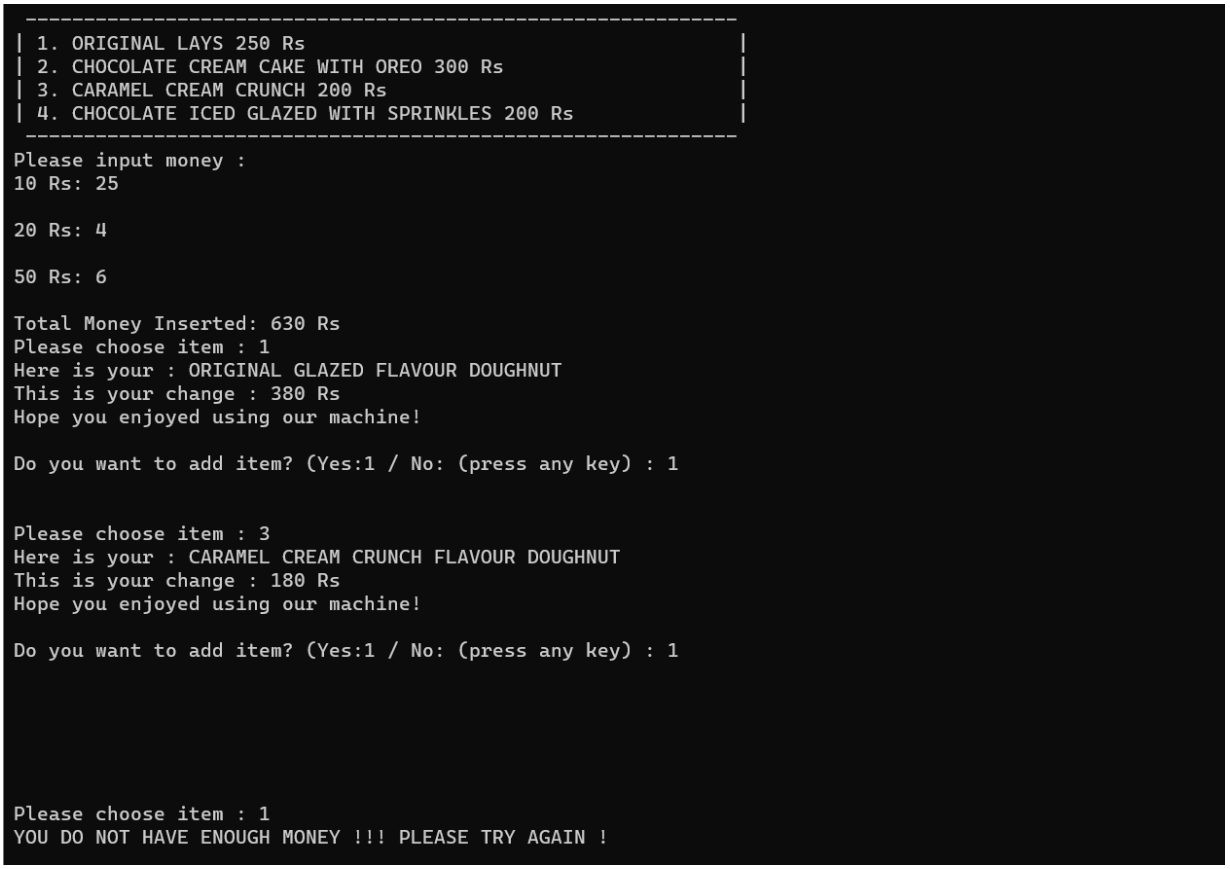
2. Example of total money inserted is not enough

```
 ------------------------------------------------------------
|     1. ORIGINAL LAYS                              250 Rs     |
|     2. CHOCOLATE CREAM CAKE WITH OREO             300 Rs     |
|     3. CARAMEL CREAM CRUNCH                       200 Rs     |
|     4. CHOCOLATE ICED GLAZED WITH SPRINKLES       200 Rs     |
 ------------------------------------------------------------
Please input money  :
10 Rs: 5

20 Rs: 2

50 Rs: 2

Total Money Inserted: 190 Rs
Please choose item  : 1
YOU DO NOT HAVE ENOUGH MONEY !!! PLEASE TRY AGAIN !
```

3. Example to continue/ stop the program

```
 ------------------------------------------------------------
|     1. ORIGINAL LAYS                              250 Rs     |
|     2. CHOCOLATE CREAM CAKE WITH OREO             300 Rs     |
|     3. CARAMEL CREAM CRUNCH                       200 Rs     |
|     4. CHOCOLATE ICED GLAZED WITH SPRINKLES       200 Rs     |
 ------------------------------------------------------------
Please input money  :
10 Rs: 5

20 Rs: 2

50 Rs: 22

Total Money Inserted: 1190 Rs
Please choose item  : 4
Here is your : CHOCOLATE ICED GLAZED WITH SPRINKLES FLAVOUR DOUGHNUT
This is your change : 990 Rs

Do you want to add item? (Yes:1 / No: (press any key) : 1


 ------------------------------------------------------------
|     1. ORIGINAL LAYS                              250 Rs     |
|     2. CHOCOLATE CREAM CAKE WITH OREO             300 Rs     |
|     3. CARAMEL CREAM CRUNCH                       200 Rs     |
|     4. CHOCOLATE ICED GLAZED WITH SPRINKLES       200 Rs     |
 ------------------------------------------------------------
Please input money  :
10 Rs:
```

4. Example to continue Spending Money:



CONCLUSION

In this project, we have applied our knowledge that we learnt from this course Assembly language to design a simple vending machine program. However, although I completed the project on time, there were still a few weaknesses in our project. Such as the program may be relatively simple without many techniques or specific assembly language I learnt. Even though the program can display the output that I desired, I still found that the program is too simple.

In conclusion, creating a vending machine in assembly language is a complex project that requires a strong understanding of both the hardware components of the vending machine and the assembly language programming language. To successfully complete the project, you will need to design and implement a program that can control the hardware and carry out the various tasks involved in vending products, such as accepting payment, checking product availability, and dispensing the product.

To create a reliable and efficient vending machine, it is important to carefully plan and test the program and hardware components. This may involve debugging and troubleshooting any issues that arise during development.

Overall, the goal of the project is to create a vending machine that is user-friendly and able to handle transactions smoothly and efficiently. By using assembly language to control the hardware, you can create a vending machine that is fast and efficient, and able to provide a seamless user experience.

REFERENCES

1. https://www.tutorialspoint.com/assembly_programming/assembly_conditions.htm