CAFÉ BILLING SYSTEM

COAL PROJECT

- MEMBERS:-

1. SHARJEEL SAFDAR GUJJAR(22K4210)
2. SUHAIB SHAIKH(22K4302)

- INTRODUCTION:-

THE CAFÉ BILLING SYSTEM IS AN ASSEMBLY LANGUAGE PROJECT DESIGNED TO STREAMLINE THE ORDERING AND BILLING PROCESS IN A RESTAURANT. THIS SYSTEM OFFERS A USERFRIENDLY INTERFACE FOR CUSTOMERS TO PLACE ORDERS AND GENERATES AN ITEMIZED BILL BASED ON THEIR SELECTIONS. THE PROJECT UTILIZES IRVINE32.INC FOR ASSEMBLY LANGUAGE PROGRAMMING.

- PROBLEM STATEMENT:-

TRADITIONAL CAFE ORDERING SYSTEMS MAY BE PRONE TO ERRORS AND INEFFICIENCIES DUE TO MANUAL PROCESSES. THERE IS A NEED FOR AN AUTOMATED SYSTEM THAT ALLOWS CUSTOMERS TO PLACE ORDERS SEAMLESSLY AND GENERATES ACCURATE BILLS.

- MOTIVATION:-

THE MOTIVATION BEHIND THIS PROJECT IS TO CREATE A RELIABLE AND EFFICIENT CAFÉ BILLING SYSTEM THAT ENHANCES THE OVERALL DINING EXPERIENCE FOR CUSTOMERS AND IMPROVES THE OPERATIONAL EFFICIENCY OF THE RESTAURANT STAFF.

- PROPOSED SOLUTION:-

THE CAFÉ BILLING SYSTEM IS DESIGNED TO ADDRESS THE CHALLENGES FACED BY TRADITIONAL CAFE ORDERING SYSTEMS. IT PROVIDES A STRUCTURED MENU WITH OPTIONS FOR BREAKFAST, LUNCH, DINNER, DRINKS, AND DESSERTS. THE SYSTEM ALLOWS CUSTOMERS TO INPUT THEIR ORDERS, AND IT CALCULATES THE TOTAL BILL BASED ON THE SELECTED ITEMS.

- METHODOLOGY:-

VISUAL STUDIO: THE PROJECT IS IMPLEMENTED USING ASSEMBLY LANGUAGE, AND VISUAL STUDIO IS THE CHOSEN INTEGRATED DEVELOPMENT ENVIRONMENT (IDE) FOR CODING AND DEBUGGING.

CODE:

```
Include Irvine32.inc

.data

    bill DWORD 0

string BYTE '                ***------------SHARJEEL GUJJAR AND SUHAIB SHEIKH CAFE ---------
----***',0dh,0ah,0

    str1 BYTE '                      *****MAIN MENU***** ',0dh,0ah

 BYTE '                                          ',0dh,0ah

 BYTE '                                          ',0dh,0ah

         BYTE '                          1. BREAK FAST       ',0dh,0ah

         BYTE '                          2. LUNCH           ',0dh,0ah

         BYTE '                          3. DINNER          ',0dh,0ah

         BYTE '                          4. DRINKS          ',0dh,0ah

         BYTE '                          5. DESSERTS         ',0dh,0ah

         BYTE '                          6. EXIT            ',0dh,0ah,0



   str2  BYTE '      1. ORDER MORE ',0dh,0ah

         BYTE '      2. GENERATE BILL ',0dh,0ah,0


   str3  BYTE '              1. Garlic Naan    = Rs 45 ',0dh,0ah

         BYTE '              2. Chapati       = Rs 15    ',0dh,0ah

         BYTE '              3. Exit             ',0dh,0ah,0


   price3 DWORD 45,15


   str4  BYTE '              1. Beef Biryani      = Rs 210 per plate  ',0dh,0ah
```

```
        BYTE '              2. Beef karahi      = Rs 380 per Kg  ',0dh,0ah
        BYTE '              3. Mix vegetable           = Rs 220 per plate  ',0dh,0ah
        BYTE '              4. Chicken Pulao     = Rs 140 per plate  ',0dh,0ah
        BYTE '              5. Exit                         ',0dh,0ah,0

price4 DWORD 210,380,220,140

str5  BYTE '           1. Paratha+Egg          = Rs 30',0dh,0ah
        BYTE '           2. Tea              = Rs 40',0dh,0ah
        BYTE '           3. Coffee           = Rs 60',0dh,0ah
        BYTE '           4. Halwa Poori        = Rs 55',0dh,0ah
        BYTE '           5. Chana        = Rs 45',0dh,0ah
        BYTE '           6.Exit                                  *',0dh,0ah,0

price5 DWORD 30,40,60,55,45

str6  BYTE  '              1. Zinger Burger        = Rs 250  ',0dh,0ah
        BYTE '              2. Pizza              = Rs 350  ',0dh,0ah
        BYTE '              3. Garlic Mayo Fries     = Rs 180  ',0dh,0ah
        BYTE '              4. Club Sandwich        = Rs 330  ',0dh,0ah
        BYTE '              5. Exit                 ',0dh,0ah,0


price6 DWORD 250,350,180,330

str7  BYTE '              1. Pepsi 500 ml      = Rs 70   ',0dh,0ah
        BYTE '              2. 7UP 500 ml        = Rs 70   ',0dh,0ah
        BYTE '              3. Fanta 500 ml      = Rs 70   ',0dh,0ah
        BYTE '              4. Dew 500 ml        = Rs 60   ',0dh,0ah
```

```
        BYTE '              5. Exit                   ',0dh,0ah,0

    price7 DWORD 70,70,70,60

    str8  BYTE '            1. Waffles          = Rs 155  ',0dh,0ah
          BYTE '            2. Hot Chocolate    = Rs 145   ',0dh,0ah
          BYTE '            3. Kheer            = Rs 75    ',0dh,0ah
          BYTE '            4. Ice-cream        = Rs 100   ',0dh,0ah
          BYTE '            5. Exit                 ',0dh,0ah,0

    price8 DWORD 155,145,75,100
    choice byte "Enter your choice: ",0
    spaces BYTE '           ',0

    Msg BYTE '     Please follow instructions correctly ',0dh,0ah,0

    Quantity BYTE '     Quantity:    ',0

    billing BYTE  '       Total Bill: Rs ',0

buff BYTE  "          ENTER NAME: ",0
order byte "          ENTER ORDER NO: ",0
buff1 byte " Name: ",0
order1 byte " Order NO: ",0
buffer BYTE 50 dup(?)
buffer1 byte 50 dup(?)
bufSize DWORD ($-buffer)
errMsg BYTE "Cannot open file",0dh,0ah,0
filename    BYTE "output.txt",0
```

```
fileHandle   HANDLE ? ; handle to output file
bytesWritten DWORD ?     ; number of bytes written

.code
main proc

    call Crlf
    call Crlf
mov eax,green
call settextcolor
mov edx,OFFSET string
    call WriteString
call crlf
call crlf
mov edx,offset order
call writestring
mov edx,offset buffer1
mov ecx,lengthof buffer1
call readstring

mov edx,offset buff
call writestring
mov edx,offset buffer
mov ecx,lengthof buffer
call readstring


;write file
mov edx, offset filename
```

```
call CreateOutputFile
mov filehandle, eax

mov eax, filehandle
mov edx, offset buffer1
mov ecx, sizeof buffer1
mov edx, offset buffer
mov ecx, sizeof buffer
call WriteToFile
mov eax, filehandle
call closeFile

;read file
;mov edx,offset filename
;call OpenInputFile
;mov filehandle, eax
;mov edx, offset buffer1
;mov ecx, sizeof buffer1
;call ReadFromFile
;mov eax,filehandle
;call closeFile
;call writestring
```

```
    L1:
call crlf
 call crlf
     mov edx,OFFSET str1
     call WriteString
 call crlf
 call crlf
     mov edx,OFFSET spaces
     call WriteString
 mov edx,offset choice
 call writestring
     call ReadDec
 mov edx,OFFSET spaces
     call WriteString
 call crlf
     call Checkerror

     cmp eax,1
     je L3
     cmp eax,2
     je L2
     cmp eax,3
     je L4
     cmp eax,4
     je L5
     cmp eax,5
     je L6
     jmp last
```

```
    L2: call lunch
        jmp L7
    L3: call breakfast
        jmp L7
    L4: call FastFood
        jmp L7
    L5: call Drinkssalad
        jmp L7
    L6: call Dessert
    L7: mov edx,OFFSET str2
        call WriteString
        mov edx,OFFSET spaces
        call WriteString
        call ReadDec
        call Checkerror1
        cmp eax,1          ; if user want to continue then jump to L1
        je L1
    last:
        call Crlf
call crlf
mov edx,offset order1
call writestring
mov edx,offset buffer1
call writestring
call crlf
call crlf

mov edx,offset buff1
```

```
call writestring
mov edx,offset buffer
call writestring
	call Crlf
call crlf
mov eax,red
	call settextcolor
	mov edx,OFFSET billing
	call WriteString
	mov eax,bill
	call WriteDec	; prints the Total bill
	call Crlf	; next line
	call Crlf
	call WaitMsg

invoke ExitProcess,0
main endp


lunch PROC

	mov edx,OFFSET str4
	call WriteString
	mov edx,OFFSET spaces
	call WriteString
mov edx,offset choice
 call writestring
	call ReadDec
call Crlf
```

```
        call Crlf
mov edx,OFFSET spaces
      call WriteString
 call crlf
        call Checkerror3
        cmp eax,1
        je L1
        cmp eax,2
        je L2
        cmp eax,3
        je L3
        cmp eax,4
        je L4
        cmp eax,5
        jmp last
 L1: mov edx,OFFSET Quantity
    call WriteString
    call ReadDec
    call Crlf
    mov ecx,eax
    mov ebx,[price4]
    L11:
      add bill,ebx          ; add price into bill
      loop L11
    jmp last
 L2: mov edx,OFFSET Quantity
    call WriteString
    call ReadDec
    call Crlf
```

```
    mov ecx,eax
    mov ebx,[price4 + 4]
    L22:
      add bill,ebx
      loop L22
    call M1
    jmp last
  L3: mov edx,OFFSET Quantity
    call WriteString
    call ReadDec
    call Crlf
    mov ecx,eax
    mov ebx,[price4 + 8]
    L33:
      add bill,ebx
      loop L33
    call M1
    jmp last
  L4: mov edx,OFFSET Quantity
    call WriteString
    call ReadDec
    call Crlf
    mov ecx,eax
    mov ebx,[price4 + 12]
    L44:
      add bill,ebx
      loop L44
    call M1
last:
```

```
ret
lunch ENDP

M1 PROC

      mov edx,OFFSET str3
      call WriteString
      mov edx,OFFSET spaces
      call WriteString
      call ReadDec
mov edx,offset choice
 call writestring
      call Checkerror2
      cmp eax,1
      je L1
      cmp eax,2
      je L2
      jmp last
   L1:
      mov ebx,[price3]    ; buuffer3 is array contains price of Naan
      mov edx,OFFSET Quantity
      call WriteString
      call ReadDec
      call Crlf
      mov ecx,eax
      L11:
        add bill,ebx
        loop L11
     jmp last
```

```
    L2:
      mov ebx,[price3 + 4]
      mov edx,OFFSET Quantity
      call WriteString
      call ReadDec
      call Crlf
      mov ecx,eax
      L22:
       add bill,ebx
       loop L22
last:
ret
M1 ENDP



breakfast PROC


      mov edx,OFFSET str5
      call WriteString
      mov edx,OFFSET spaces
      call WriteString
mov edx,offset choice
 call writestring
      call ReadDec
      call Crlf
 mov edx,OFFSET spaces
      call WriteString
call crlf
      call Checkerror3      ; check for error
```

```
    cmp eax,1
    je L1
    cmp eax,2
    je L2
    cmp eax,3
    je L3
    cmp eax,4
    je L4
    cmp eax,5
    jmp last

L1: mov edx,OFFSET Quantity
   call WriteString
   call ReadDec
   mov ecx,eax
   mov ebx,[price5]
   L11:
     add bill,ebx
     loop L11
   jmp last
L2: mov edx,OFFSET Quantity
   call WriteString
   call ReadDec
   mov ecx,eax
   mov ebx,[price5 + 4]
   L22:
      add bill,ebx
      loop L22
   jmp last
```

```
L3: mov edx,OFFSET Quantity
    call WriteString
    call ReadDec
    call Crlf
    mov ecx,eax
    mov ebx,[price5 + 8]
    L33:
       add bill,ebx
       loop L33
    jmp last
L4: mov edx,OFFSET Quantity
    call WriteString
    call ReadDec
    call Crlf
    mov ecx,eax
    mov ebx,[price5 + 12]
    L44:
       add bill,ebx
       loop L44
last:

ret
breakfast ENDP

FastFood PROC

       mov edx,OFFSET str6
       call WriteString
       mov edx,OFFSET spaces
```

```
        call WriteString
mov edx,offset choice
  call writestring
        call ReadDec
call Crlf
        call Crlf
mov edx,OFFSET spaces
      call WriteString
  call crlf
        call Checkerror3
        cmp eax,1
        je L1
        cmp eax,2
        je L2
        cmp eax,3
        je L3
        cmp eax,4
        je L4
        cmp eax,5
        jmp last
 L1: mov edx,OFFSET Quantity
    call WriteString
    call ReadDec
    call Crlf
    mov ecx,eax
    mov ebx,[price6]
    L11:
      add bill,ebx
      loop L11
```

```
    jmp last
L2: mov edx,OFFSET Quantity
    call WriteString
    call ReadDec
    call Crlf
    mov ecx,eax
    mov ebx,[price6 + 4]
    L22:
      add bill,ebx
      loop L22
    jmp last
L3: mov edx,OFFSET Quantity
    call WriteString
    call ReadDec
    call Crlf
    mov ecx,eax
    mov ebx,[price6 + 8]
    L33:
      add bill,ebx
      loop L33
    jmp last
L4: mov edx,OFFSET Quantity
    call WriteString
    call ReadDec
    call Crlf
    mov ecx,eax
    mov ebx,[price6 + 12]
    L44:
      add bill,ebx
```

```
        loop L44
last:
ret
FastFood ENDP

Drinkssalad PROC

        mov edx,OFFSET str7
        call WriteString
        mov edx,OFFSET spaces
        call WriteString
mov edx,offset choice
 call writestring
        call ReadDec
call Crlf
        call Crlf
mov edx,OFFSET spaces
     call WriteString
 call crlf
        call Checkerror3
        cmp eax,1
        je L1
        cmp eax,2
        je L2
        cmp eax,3
        je L3
        cmp eax,4
        je L4
        cmp eax,5
```

```
        jmp last
L1: mov edx,OFFSET Quantity
    call WriteString
    call ReadDec
    call Crlf
    mov ecx,eax
    mov ebx,[price7]
    L11:
      add bill,ebx
      loop L11
    jmp last
L2: mov edx,OFFSET Quantity
    call WriteString
    call ReadDec
    call Crlf
    mov ecx,eax
    mov ebx,[price7 + 4]
    L22:
       add bill,ebx
       loop L22
    jmp last
L3: mov edx,OFFSET Quantity
    call WriteString
    call ReadDec
    call Crlf
    mov ecx,eax
    mov ebx,[price7 + 8]
    L33:
       add bill,ebx
```

```
        loop L33
    jmp last
L4: mov edx,OFFSET Quantity
    call WriteString
    call ReadDec
    call Crlf
    mov ecx,eax
    mov ebx,[price7 + 12]
    L44:
        add bill,ebx
        loop L44
last:
ret
Drinkssalad ENDP

Dessert PROC

        mov edx,OFFSET str8
        call WriteString
        mov edx,OFFSET spaces
        call WriteString
        call ReadDec
call Crlf
        call Crlf
mov edx,OFFSET spaces
      call WriteString
 call crlf
        call Checkerror3
        cmp eax,1
```

```
        je L1
        cmp eax,2
        je L2
        cmp eax,3
        je L3
        cmp eax,4
        je L4
        cmp eax,5
        jmp last
L1: mov edx,OFFSET Quantity
    call WriteString
    call ReadDec
    call Crlf
    mov ecx,eax
    mov ebx,[price8]
    L11:
      add bill,ebx
      loop L11
    jmp last
L2: mov edx,OFFSET Quantity
    call WriteString
    call ReadDec
    call Crlf
    mov ecx,eax
    mov ebx,[price8 + 4]
    L22:
       add bill,ebx
       loop L22
    jmp last
```

```
L3: mov edx,OFFSET Quantity
    call WriteString
    call ReadDec
    mov ecx,eax
    call Crlf
    mov ebx,[price8 + 8]
    L33:
       add bill,ebx
       loop L33
    jmp last
L4: mov edx,OFFSET Quantity
    call WriteString
    call ReadDec
    call Crlf
    mov ecx,eax
    mov ebx,[price8 + 12]
    L44:
       add bill,ebx
       loop L44

last:
ret
Dessert ENDP

Checkerror PROC

L1:
  cmp eax,1
  jl L2
```

```
  cmp eax,6
  jg L2
  jmp last

L2:
  mov edx, OFFSET Msg
  call WriteString
  call ReadDec
  jmp L1
last:
ret
Checkerror ENDP

Checkerror1 PROC

L1:
  cmp eax,1
  jl L2
  cmp eax,2
  jg L2
  jmp last

L2:
  mov edx, OFFSET Msg
  call WriteString
  call ReadDec
  jmp L1
last:
ret
```

```
Checkerror1 ENDP

Checkerror2 PROC

L1:
  cmp eax,1
  jl L2
  cmp eax,3
  jg L2
  jmp last

L2:
  mov edx, OFFSET Msg
  call WriteString
  call ReadDec
  jmp L1
last:
ret
Checkerror2 ENDP

Checkerror3 PROC

L1:
  cmp eax,1
  jl L2
  cmp eax,5
  jg L2
  jmp last
```

```
L2:
  mov edx, OFFSET Msg
  call WriteString
  call ReadDec
  jmp L1
last:
ret
Checkerror3 ENDP

end main
```

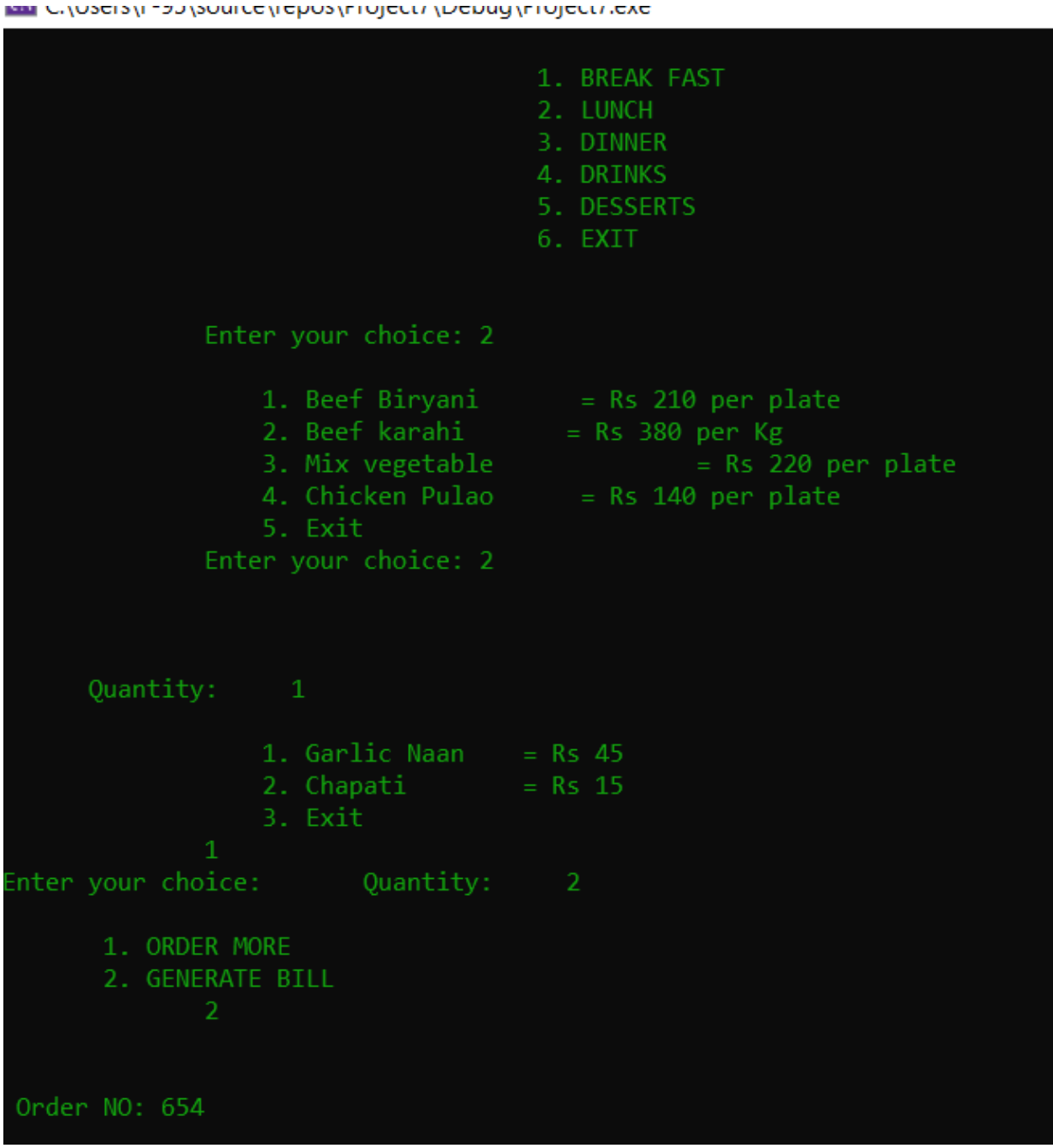
RESULT EXPLANATION: THE SYSTEM PROMPTS THE USER TO INPUT THEIR NAME, ORDER NUMBER, AND THEN GUIDES THEM THROUGH THE MENU OPTIONS. THE USER CAN CHOOSE ITEMS FROM DIFFERENT CATEGORIES, SPECIFY QUANTITIES, AND GENERATE A DETAILED BILL. THE SYSTEM HANDLES ERRORS AND ENSURES A SMOOTH ORDERING PROCESS.

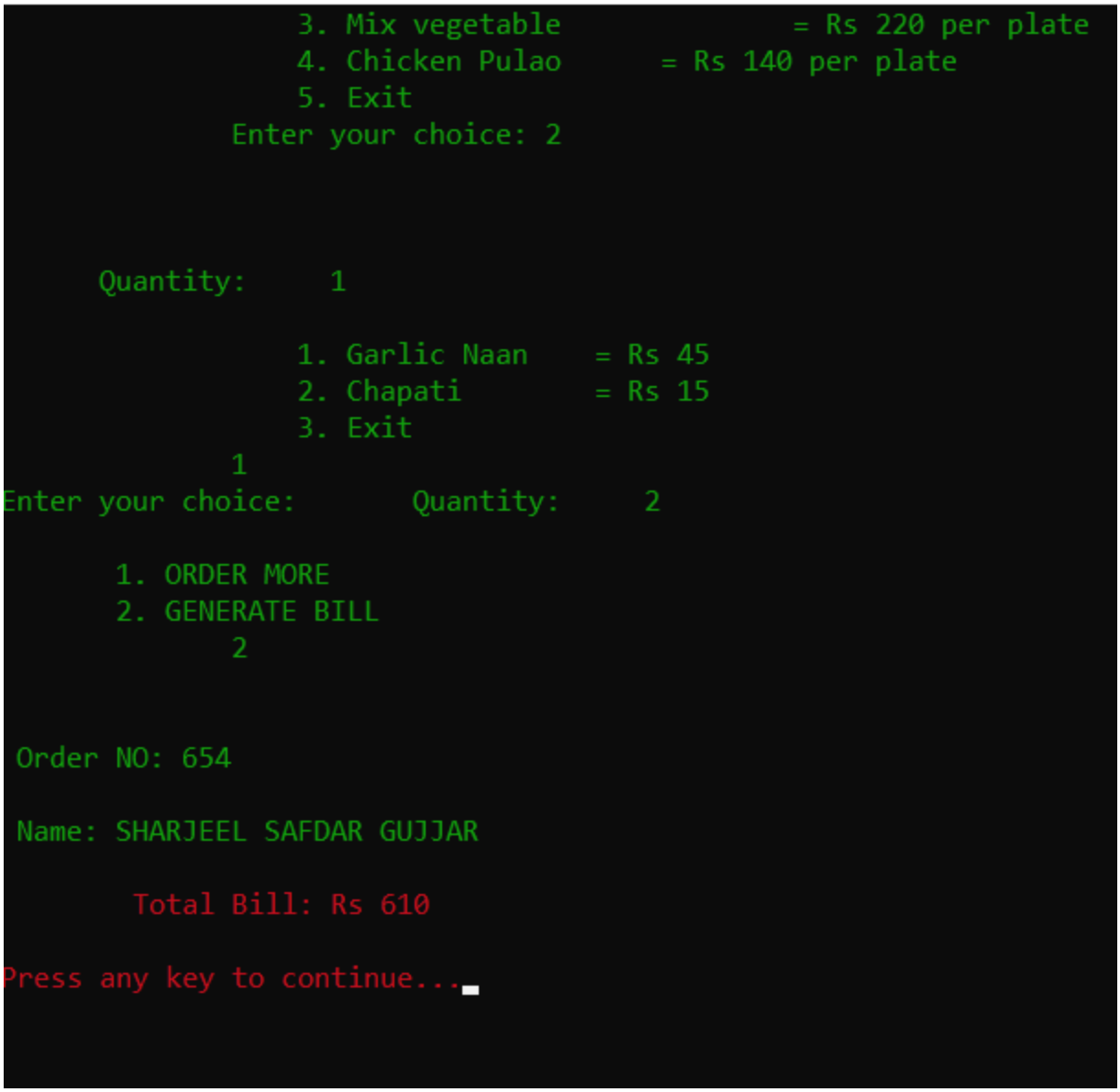
SCREEN SHOTS:

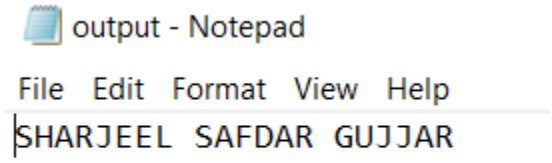
C:\Users\F-95\source\repos\Project7\Debug\Project7.exe

```
                ***------------SHARJEEL GUJJAR AND SUHAIB SHEIKH CAFE -------------***


        ENTER ORDER NO: 654
        ENTER NAME: SHARJEEL SAFDAR GUJJAR


                        *****MAIN MENU*****


                                1. BREAK FAST
                                2. LUNCH
                                3. DINNER
                                4. DRINKS
                                5. DESSERTS
                                6. EXIT


         Enter your choice: 4

             1. Pepsi 500 ml       = Rs 70
             2. 7UP 500 ml         = Rs 70
             3. Fanta 500 ml       = Rs 70
             4. Dew 500 ml         = Rs 60
             5. Exit
         Enter your choice: 1



    Quantity:     2

     1. ORDER MORE
     2. GENERATE BILL
            1
```

```
        Enter your choice: 1


Quantity:     2

 1. ORDER MORE
 2. GENERATE BILL
        1


                        *****MAIN MENU*****


                                1. BREAK FAST
                                2. LUNCH
                                3. DINNER
                                4. DRINKS
                                5. DESSERTS
                                6. EXIT


        Enter your choice: 2

            1. Beef Biryani       = Rs 210 per plate
            2. Beef karahi       = Rs 380 per Kg
            3. Mix vegetable              = Rs 220 per plate
            4. Chicken Pulao      = Rs 140 per plate
            5. Exit
        Enter your choice: 2


Quantity:     1
```


1. BREAK FAST
2. LUNCH
3. DINNER
4. DRINKS
5. DESSERTS
6. EXIT
Enter your choice: 2
1. Beef Biryani = Rs 210 per plate
2. Beef karahi = Rs 380 per Kg
3. Mix vegetable = Rs 220 per plate
4. Chicken Pulao = Rs 140 per plate
5. Exit
Enter your choice: 2
Quantity: 1
1. Garlic Naan = Rs 45
2. Chapati = Rs 15
3. Exit
1
Enter your choice: Quantity: 2
1. ORDER MORE
2. GENERATE BILL
2
Order NO: 654





- BENEFITS/ADVANTAGES/DISADVANTAGES:-

BENEFITS:-

1. AUTOMATION: AUTOMATES THE ORDERING AND BILLING PROCESS, REDUCING MANUAL ERRORS.

2. EFFICIENCY: STREAMLINES CAFE OPERATIONS AND IMPROVES ORDER PROCESSING SPEED.

3. USER-FRIENDLY: PROVIDES A STRAIGHTFORWARD INTERFACE FOR CUSTOMERS TO PLACE ORDERS.

4. ACCURACY: CALCULATES THE TOTAL BILL ACCURATELY BASED ON USER SELECTIONS.

ADVANTAGES:-

1. COST SAVINGS: REDUCES THE NEED FOR MANUAL ORDER PROCESSING, SAVING TIME AND RESOURCES.

2. CUSTOMER SATISFACTION: ENHANCES THE OVERALL DINING EXPERIENCE BY PROVIDING A CONVENIENT ORDERING SYSTEM.

3. FLEXIBILITY: CAN BE CUSTOMIZED AND EXTENDED TO INCLUDE ADDITIONAL FEATURES AND MENU ITEMS.

DISADVANTAGES:-

1. TECHNICAL DEPENDENCY: RELIES ON ASSEMBLY LANGUAGE, WHICH MAY HAVE A STEEPER LEARNING CURVE FOR SOME DEVELOPERS.

2. LIMITED FEATURES: THE SYSTEM MAY LACK ADVANCED FEATURES PRESENT IN MORE MODERN CAFÉ SYSTEMS.

- CONCLUSION:-

THE CAFÉ BILLING SYSTEM IS A SIGNIFICANT STEP TOWARDS IMPROVING THE EFFICIENCY AND ACCURACY OF RESTAURANT OPERATIONS. BY AUTOMATING THE ORDERING AND BILLING PROCESSES, THE SYSTEM CONTRIBUTES TO A SMOOTHER DINING EXPERIENCE FOR BOTH CUSTOMERS AND CAFÉ STAFF.

- REFERENCE:-

THIS PROJECT WAS THE OUR SECOND SEMESTER OBJECT-ORIENTED PROGRAMMING IN C++ WE CONVERTED THIS INTO ASSEMBLY LANGUAGE WITH THE HELP OF DIFFERENT RESOURCES AND RESEARCH.

- CONTRIBUTION:-

BOTH MEMBERS RESEARCHED AND DESIGNED THIS PROJECT TOGATHER USING RESOURCES WORK AND LOGIC WERE COMBINED IN COMBINED SESSION FOR THIS PROJECT IN GROUP SOME IDEAS ARE RELATED TO SUHAIB THINKING AND REMAINING OF SHARJEEL.OVERALL EACH MEMBER CONTRIBUTE EQUALLY.