Lab: 8

Fahid Imran

Roll No: 23i-0061

COAL

Instructor

Mr. Sulaman Saboor

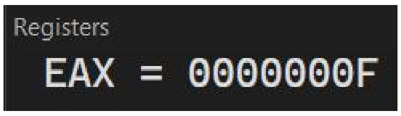
Fast NUCES Islamabad Campus

Task1:

Code:

```
.386
.model flat, stdcall
.stack 4096


.data
sum_var dword 0

.code

SUM1 PROC
  mov eax, 0
  mov ebx, 0
  mov ecx, 5
  pop ebx
  label1:
  pop ebx
  add eax, ebx
  loop label1

  ret
SUM1 endp

main PROC
  push 1
  push 2
  push 3
  push 4
  push 5

  call SUM1
  mov sum_var, eax

main endp
end main
```

Output:



```
Registers
  EAX = 0000000F
```

Task2:

Code:

```
.386
.model flat, stdcall
.stack 4096



.data
total dword 0
Average dword 0
Max_M dword 0
Min_M dword 0
var1 dword 0
marks dword 12,23,14,54,67,89,34, 56, 89,34,23
size_of dword ?
.code

getTotal PROC
  mov eax, 0
  mov eax, 0
  pop ebx
  pop esi
  pop ecx
  push ebx

  label1:

  add eax, dword ptr [esi]
  add esi, 4

  loop label1



  ret
```

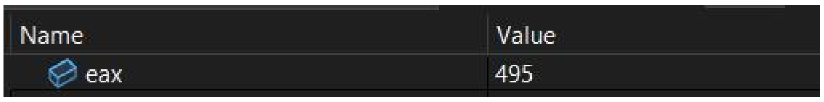
```
getTotal endp

getAverage PROC
  mov eax, 0
  mov eax, 0
  pop ebx
  mov var1, ebx
  call getTotal
  push var1
  mov edx, 0
  mov ebx, size_of
  div ebx
  ret
getAverage endp

getMin PROC
  mov eax, 0
  pop ebx
  pop esi
  pop ecx
  push ebx

  mov eax, dword ptr [esi]
  add esi, 4
  dec ecx

  label1:

  cmp eax, dword ptr [esi]
  jg jmp1
  jmp end1
  jmp1:
        mov eax ,dword ptr [esi]
  end1:
  add esi, 4

  loop label1


  ret
getMin endp
```

```
getMax PROC
  mov eax, 0
  pop ebx
  pop esi
  pop ecx
  push ebx

  mov eax, dword ptr [esi]
  add esi, 4
  dec ecx

  label1:

  cmp dword ptr [esi], eax
  jg jmp1
  jmp end1
  jmp1:
        mov eax ,dword ptr [esi]
  end1:
  add esi, 4

  loop label1


  ret
getMax endp

main PROC
  push lengthof marks
  mov size_of, lengthof marks
  push offset marks


  call getTotal
  mov total, eax

  push lengthof marks
  push offset marks
  call getAverage
  mov average, eax

  push lengthof marks
```

```
        push offset marks
        call getMin
        mov min_m, eax

        push lengthof marks
        push offset marks
        call getMax
        mov max_m, eax

    main endp
end main
```

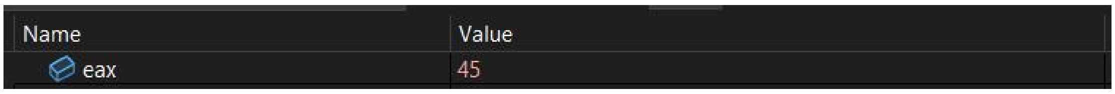
Output:

Total:



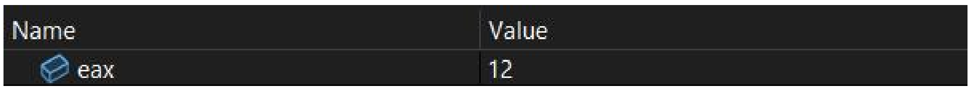
```
Name        Value
  eax       495
```

Average:



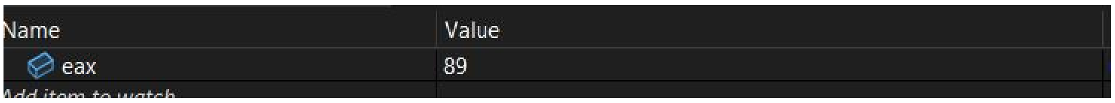
```
Name        Value
  eax       45
```

Min:



```
Name        Value
  eax       12
```

Max:



```
Name        Value
  eax       89
Add item to watch
```

Task3:

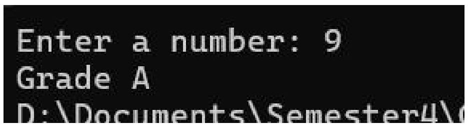
Code:

```
include Irvine32.inc
.386
.model flat, stdcall
.stack 4096


.data


msg Byte 'Enter a number: ', 0
```

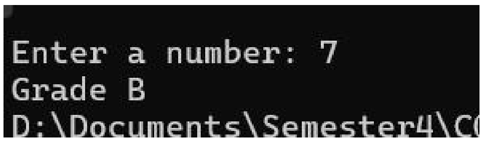
```
msg1 Byte 'Grade A', 0
msg2 Byte 'Grade B', 0
msg3 Byte 'Grade C', 0
msg4 Byte 'Grade D', 0


.code

getGrade PROC
  pop ebx
  pop eax
  push ebx

  cmp eax, 5
  jl jmp1
  jmp end1
  jmp1:
        mov edx ,offset msg4
        call writestring
        ret

  end1:
  cmp eax, 7
  jl jmp2
  jmp end2
  jmp2:
        mov edx ,offset msg3
        call writestring
        ret

  end2:
  cmp eax, 9
  jl jmp3
  jmp end3
  jmp3:
        mov edx ,offset msg2
        call writestring
        ret
  end3:
        mov edx ,offset msg1
        call writestring
        ret
```

```
    getGrade endp



    main PROC
      mov edx , offset msg
      call writestring
      call readint

      push eax
      call getGrade
    exit
    main endp
end main
```

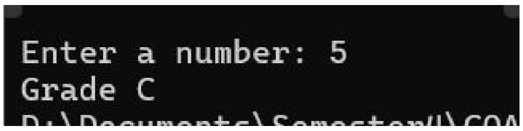
Output:



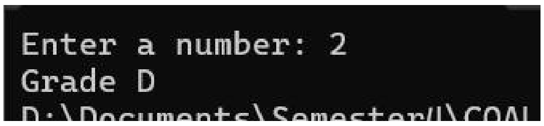
```
Enter a number: 9
Grade A
D:\Documents\Semester4\
```



```
Enter a number: 7
Grade B
D:\Documents\Semester4\C
```



```
Enter a number: 5
Grade C
D:\Documents\Semester4\COA
```



```
Enter a number: 2
Grade D
D:\Documents\Semester4\COAL
```

Task4:

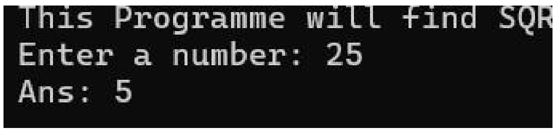
Code:

```
include Irvine32.inc
.386
.model flat, stdcall
.stack 4096


.data
```

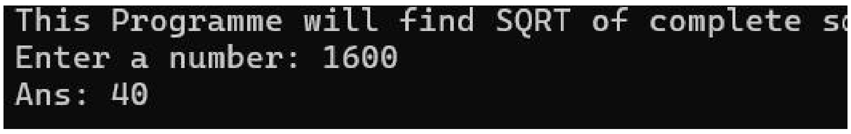
```
num dword ?
msg Byte 'This Programme will find SQRT of complete square numbers
from 1 to 2500. ', 0
str1 Byte 'It is not complete sqrt or it is not in range (1 to
2500).' , 0
msg1 Byte 'Enter a number: ', 0
msg2 Byte 'Ans: ', 0



.code

Sq_root PROC
  pop edx
  pop ebx
  mov num, ebx
  push edx

  mov ecx, 50
  mov ebx, 0

  label1:

        mov eax, ebx
        mul ebx

        cmp eax, num
        je go1
        jmp go2
        go1:
              mov edx , offset msg2
              call writestring
              mov eax, ebx
              call writedec
              call crlf
              ret
        go2:
        inc ebx
  loop label1
  mov edx , offset str1
  call writestring
```

```
      ret

    Sq_root endp

    main PROC
      mov edx , offset msg
      call writestring
      call crlf
      mov edx , offset msg1
      call writestring
      call readint

      push eax
      call Sq_root

    exit
    main endp
end main
```

Output:



```
This Programme will find SQR
Enter a number: 25
Ans: 5
```



```
This Programme will find SQRT of complete sq
Enter a number: 1600
Ans: 40
```

Task5:

Code:

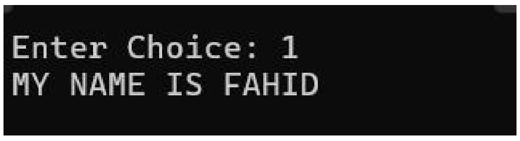
```
include Irvine32.inc
.386
.model flat, stdcall
.stack 4096


.data

size_of dword ?
msg Byte 'Enter Choice: ', 0
str1 Byte 'my name is fahid' , 0
msg1 Byte 'Enter a number: ', 0
```

```
msg2 Byte 'Ans: ', 0



.code

Capitalize PROC
  pop eax
  pop esi
  push eax


  mov ecx, size_of
  label1:
        mov al, byte ptr [esi]
        cmp al, 061h
        jge jump1
        jmp endqw
        jump1:
        cmp al, 07ah
        jle jump2
        jmp endqw

        jump2:
              and al, 11011111b
              mov byte ptr [esi], al
        endqw:
        inc esi

  loop label1
  ret
Capitalize endp

Print PROC
  pop eax
  pop edx
  push eax

  call writestring
  call crlf

  ret
```

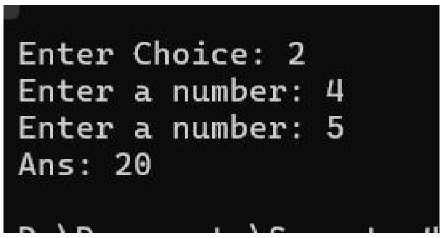
```
Print endp

mulNum PROC
  pop edx
  pop eax
  pop ebx
  push edx

  mul ebx

  ret

mulNum endp

main PROC
  mov edx , offset msg
  call writestring
  call readint

  cmp eax, 1
  je go1
  jmp go2

  go1:

  mov size_of, lengthof str1
  push offset str1
  call Capitalize
  push offset str1
  call Print
  jmp endyu
  go2:
  cmp eax, 2
  je go3
  jmp endyu
  go3:
  mov edx, offset msg1
  call writestring
  call readint
  push eax
```

```
        mov edx, offset msg1
        call writestring
        call readint
        push eax

        call mulNum

        mov edx, offset msg2
        call writestring
        call writedec
        call crlf

        endyu:

    exit
    main endp
end main
```

Output:



```
Enter Choice: 1
MY NAME IS FAHID
```



```
Enter Choice: 2
Enter a number: 4
Enter a number: 5
Ans: 20
```