Lab: 11

Fahid Imran

Roll No: 23i-0061

COAL

Instructor

Mr. Sulaman Saboor

Fast NUCES Islamabad

Campus

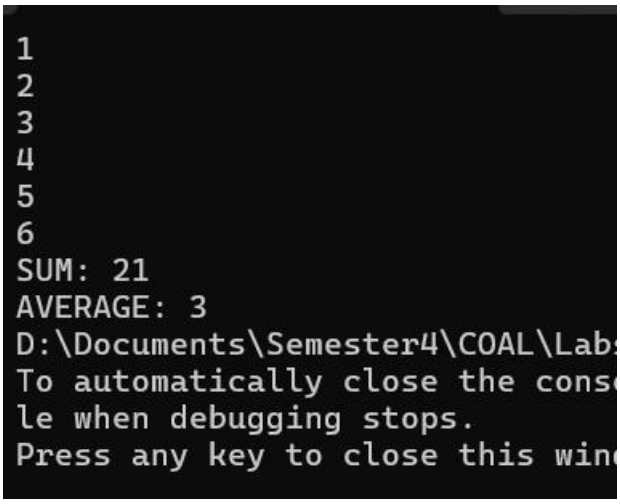
Task1:

Code:

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

.data
  arr dword 6 dup (?)
  msg1 byte "SUM: ",0
  msg2 byte "AVERAGE: ",0
.code

getSUM PROC
  pop ebp
  pop ecx
  pop ebx
  push ebp
  push ecx
  push ebx

  mov esi , 0
  mov eax, 0
  label1:
        add eax, dword ptr [ebx + esi * 4]
        add esi, 1
  loop label1
  pop ebx
  pop ecx
  ret
getSUM endp


getAVERAGE PROC
  pop ebp
  pop ecx
  pop ebx
  push ebp
  push ebx
  push ecx
```

```
  call getSUM
  mov edx , 0
  div ecx
  ret
getAVERAGE endp

main PROC
  mov ebx, offset arr
  mov esi, 0
  mov ecx, 6

  label2:
        call readint
        mov dword ptr [ebx + esi* type arr], eax
        add esi, 1
  loop label2

  mov ebx, offset arr
  mov ecx, lengthof arr
  push ebx
  push ecx
  call getSUM
  mov edx, offset msg1
  call writestring
  call writedec
  call crlf

  mov ebx, offset arr
  mov ecx, lengthof arr
  push ebx
  push ecx
  call getAVERAGE
  mov edx, offset msg2
  call writestring
  call writedec

exit
main endp
end main
```

Output:



Task2:

Code:

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

.data
  arr dword 6 dup (?)
  msg1 byte "Enter a number: ",0
  msg2 byte "BINARY: ",0
.code


getBINARY PROC
  pop ebp
  pop eax
  mov edx, offset msg2
  call writestring
  call WriteBin
  push eax
  push ebp
```

```
  ret
getBINARY endp

main PROC

  mov edx, offset msg1
  call writestring
  call readint
  push eax
  call getBINARY
  call crlf




exit
main endp
end main
```

Output:

```
Enter a number: 1256373657
BINARY: 0100 1010 1110 0010 1011 1101 1001 1001
```

Task3:

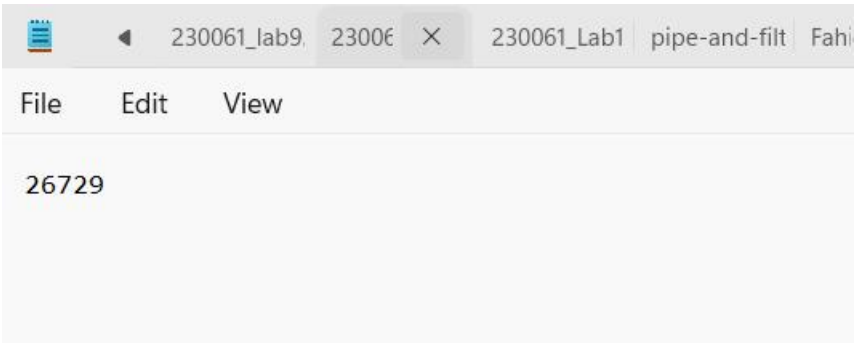
Code:

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

.data
  buffsize = 6
  data byte buffsize dup (?)
  filehandler handle ?
  msg1 byte "Enter a number: ",0
  filename byte "info.txt",0
```

```
.code


writeDataToFile PROC
  push edx
  push ebx
  push ecx


  mov edx, offset filename
  call createoutputfile
  mov filehandler, eax

  pop ecx
  pop edx
  push edx
  push ecx

  call crlf
  call writetofile
  mov eax, filehandler
  call closefile

  pop edx
  pop ecx
  pop ebx
  ret
writeDataToFile endp

main PROC

  mov ebx, offset data
  mov esi , 0
  mov ecx, 5
  label2:
        mov edx, offset msg1
        call writestring
        call readint
        add al, 48
        mov byte ptr [ebx + esi * type data], al
        add esi, 1
  loop label2
```

```
  mov ebx, offset data
  mov ecx, 5
  call writeDataToFile

exit
main endp
end main
```

Output:



Task4:

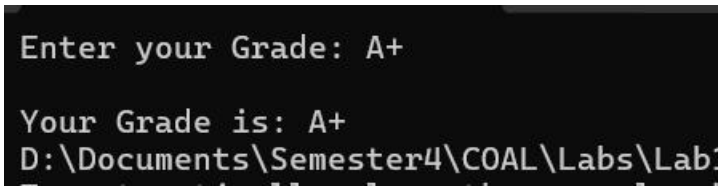
Code:

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

.data
  buffsize = 5
  data byte buffsize dup (?)
  filehandler handle ?
  msg1 byte "Enter your Grade: ",0
  msg2 byte "Your Grade is: ",0
  filename byte "grade.txt",0
  len dword ?
.code
```

```
writeDataToFile PROC
  push edx
  push ebx
  push ecx


  mov edx, offset filename
  call createoutputfile
  mov filehandler, eax

  pop ecx
  pop edx
  push edx
  push ecx

  call crlf
  call writetofile
  mov eax, filehandler
  call closefile

  pop edx
  pop ecx
  pop ebx
  ret
writeDataToFile endp


readDataFromFile PROC
  push edx
  push ebx
  push ecx


  mov edx, offset filename
  call openinputfile
  mov filehandler, eax

  pop ecx
  pop edx
  push edx
  push ecx
```

```
  call readfromfile
  mov len, eax
  mov eax, filehandler
  call closefile


  pop edx
  pop ecx
  pop ebx
  ret
readDataFromFile endp

main PROC

  mov edx, offset msg1
  call writestring
  mov edx, offset data
  mov ecx, buffsize
  call readstring


  mov ebx, offset data
  mov ecx, 1
  call writeDataToFile


  mov ebx, offset data
  mov ecx, 1
  call readDataFromFile

  mov edx, offset msg2
  call writestring
  mov edx, offset data
  call writestring

exit
main endp
end main
```

Output:



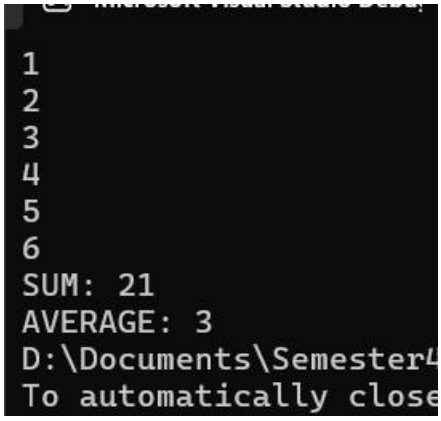
Task5:

Code:

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

.data
  arr dword 6 dup (?)
  msg1 byte "SUM: ",0
  msg2 byte "AVERAGE: ",0
.code

getSUM PROC
  push ebx
  push ecx
  mov esi , 0
  mov eax, 0
  label1:
        add eax, dword ptr [ebx + esi * 4]
        add esi, 1
  loop label1

  pop ecx
  pop ebx
  ret
getSUM endp


getAVERAGE PROC
  push ebx
  push ecx
```

```
  call getSUM
  mov edx , 0
  div ecx

  pop ecx
  pop ebx
  ret
getAVERAGE endp

main PROC
  mov ebx, offset arr
  mov esi, 0
  mov ecx, 6

  label2:
        call readint
        mov dword ptr [ebx + esi* type arr], eax
        add esi, 1
  loop label2
  mov ebx, offset arr
  mov ecx, lengthof arr
  call getSUM
  mov edx, offset msg1
  call writestring
  call writedec
  call crlf
  call getAVERAGE
  mov edx, offset msg2
  call writestring
  call writedec

exit
main endp
end main
```

Output:


1
2
3
4
5
6
SUM: 21
AVERAGE: 3
D:\Documents\Semester
To automatically close