GROUP 14

NAME	REGISTRATION NUMBER
MUGUMBYA BENON	21/U/05126/PS
KAKAI SHANICE NORAH PANDE	21/U/12393/PS
OLOO WILLIAM	21/U/19370/PS
SAHRA AHMED	21/X/20199/PS
AHUMUZA DERRICK	21/U/06270/PS
KAREN KIFUKO	21/U/14336/PS
BOGERA HENRY	21/U/16390/EVE

1. jmp   next

NUMBER 2

```
section     .text

global   main :

main  :      ;tell linker entry point

          ;adding 20 and 13

           mov eax, ‘20’ ;move 20 to eax

           sub eax,  ‘0’ ; convert value in eax to decimal

           mov ebx,  ‘13’;store 13 in ebx

           sub ebx, ‘0’ ; convert 13 to decimal

           add ebx,  eax  ;add the value in eax to ebx  and store it in ebx

         mov num, ebx   ; store the value of ebx in     num

          int    0x80   ; call kernel


            mov eax,   ‘82’

            sub eax,  ‘0’  ; convert 82 to decimal
            add eax,   num  ; adds the value of num to eax and stores
                                                        value in eax

             mov  ebx,  ‘3’  ; store ‘3’ in ebx

             sub  ebx, ‘0’  ;converts ‘3’ to decimal

             div  ebx        ;    divides value in eax by 3 and stores
result in eax

           add   eax,  ‘0’  ; converts the number from decimal to a
                                                           string

            mov   [res],   eax

            mov  ecx,   msg

            mov  edx,  len

            mov   ebx,  1; file descriptor(stdout)

            mov   eax,  4  ; system call  number (sys_write)

             int     0x80


             mov eax,   1 ;file descriptor_(stdout)
             mov  ebx,  4  ; system call number (sys_write)
             mov  ecx, res ; store the value of res in ecx
             mov edx,  1 ; stores the length of res in edx
             int    0x80  ; call kernel
```

```
section    .data
      msg  db ‘The average of 20,13 and 82: ‘,0xA,0xD
      len   equ $-  msg ; get size of msg

segment     .bss
       res  resb  1 ; reserve 1 byte for res
       num resb 1 ; reserve 1 byte for num
```

Number 3

mov eax, a ; This moves the binary representation of a into the eax register

mov ebx, b ; This moves the value b of its binary representation to the register ebx

xo eax, x ; The xor operation is applied on the x and the value of eax which is now a and the value stored in eax. The xor operation yields a 1 when the matching bits are different and a 0 when the matching bits are the same

xor ebx, y ; xor operation performed on the binary value of y and ebx contents (which is now b) and the value stored in ebx.

or eax, ebx ; or operation applied on ebx and eax where the or operation yields 1 when one or all the matching bits contains 1 and yields a 0 when all the values in the matching bits are 0
The result is stored in eax.

jnz L2 ; instructs the assembler to jump (change sequence of compiling instructions) and read the instructions with label L2 if the result of ‘or eax, ebx’ is not a zero , if the result is a zero then the assembler will read the instructions in L1 first.

Jmp L3 ;instructs the assembler to skip the instructions in L2and read the instructions in L3.

Number 4

Advantages of assembly language

- It enables a programmer to understand interface of programs with OS, processor and BIOS
- It enables the programmer understand how a program access external devices. Other advantages of using assembly language are
- It requires less memory and execution time
- It allows hardware-specific complex jobs in an easier way
- It is suitable for time-critical jobs

Number 5

- It is hard to debug and verify code of chances of error possibilities.
- Code is hard to maintain because it unstructured
- You need to know the different code representation for each platform which is tedious to programmers.
- High level languages contain intrinsic functions that are easier to use than assembly language.
- Writing code is much more time consuming and therefore time wasting