UTM UNIVERSITI TEKNOLOGI MALAYSIA

Sekolah Pendidikan Profesional dan Pendidikan Berterusan (SPACE)

**FINAL EXAMINATION / *PEPERIKSAAN AKHIR***
**SEMESTER II – SESSION 2018/ 2019**
**PROGRAM KERJASAMA**

WC

| | | |
|---|---|---|
| COURSE CODE<br>*KOD KURSUS* | : | DDWC 1223 |
| COURSE NAME<br>*NAMA KURSUS* | : | COMPUTER ORGANIZATION & ASSEMBLY LANGUAGE<br>*ORGANISASI KOMPUTER & BAHASA HIMPUNAN* |
| YEAR / PROGRAMME<br>*TAHUN / PROGRAM* | : | 1DDWC / 1 DDWZ |
| DURATION<br>*TEMPOH* | : | 2 HOURS 30 MINUTES |
| DATE<br>*TARIKH* | : | APRIL 2019 |

INSTRUCTION/*ARAHAN* :

Answer **ALL** questions in the spaces provided in this question paper.

*Jawab **SEMUA** soalan di ruang yang disediakan dalam kertas soalan ini.*

(You are required to write your name and your lecturer's name on your answer script)
*( Pelajar dikehendaki tuliskan nama dan nama pensyarah pada skrip jawapan )*

| | | |
|---|---|---|
| NAME / *NAMA* | : | ..... |
| I.C NO. / *NO. K/PENGENALAN* | : | ..... |
| YEAR / COURSE<br>*TAHUN / KURSUS* | : | ..... |
| COLLEGE<br>*KOLEJ* | : | ..... |
| LECTURER'S NAME<br>*NAMA PENSYARAH* | : | ..... |

This examination paper consists of ...19... pages including the cover
*Kertas soalan ini mengandungi .....19..... muka surat termasuk kulit hadapan*

## PUSAT PROGRAM KERJASAMA

# PETIKAN DARIPADA PERATURAN AKADEMIK
# ARAHAN AM - PENYELEWENGAN AKADEMIK

## 1. SALAH LAKU SEMASA PEPERIKSAAN

1.1 Pelajar tidak boleh melakukan mana-mana salah laku peperiksaan seperti berikut :-

1.1.1 memberi dan/atau menerima dan/atau memiliki sebarang maklumat dalam bentuk elektronik, bercetak atau apa jua bentuk lain yang tidak dibenarkan semasa berlangsungnya peperiksaan sama ada di dalam atau di luar Dewan Peperiksaan melainkan dengan kebenaran Ketua Pengawas; atau

1.1.2 menggunakan makluman yang diperolehi seperti di atas bagi tujuan menjawab soalan peperiksaan; atau

1.1.3 menipu atau cuba untuk menipu atau berkelakuan mengikut cara yang boleh ditafsirkan sebagai menipu semasa berlangsungnya peperiksaan; atau

1.1.4 lain-lain salah laku yang ditetapkan oleh Universiti (seperti membuat bising, mengganggu pelajar lain, mengganggu Pengawas menjalankan tugasnya).

## 2. HUKUMAN SALAH LAKU PEPERIKSAAN

2.1 Sekiranya pelajar didapati telah melakukan pelanggaran mana-mana peraturan peperiksaan ini, setelah diperakukan oleh Jawatankuasa Peperiksaan Fakulti dan disabitkan kesalahannya, Senat boleh mengambil tindakan dari mana-mana satu yang berikut :-

2.1.1 memberi markah SIFAR (0) bagi keseluruhan keputusan peperiksaan kursus yang berkenaan (termasuk kerja kursus); atau

2.1.2 memberi markah SIFAR (0) bagi semua kursus yang didaftarkan pada semester tersebut.

2.2 Jawatankuasa Akademik Fakulti boleh mencadangkan untuk diambil tindakan tatatertib mengikut peruntukan Akta Universiti dan Kolej Universiti, 1971, Kaedah-kaedah Universiti Teknologi Malaysia (Tatatertib Pelajar-pelajar), 1999 bergantung kepada tahap kesalahan yang dilakukan oleh pelajar.

2.3 Pelajar yang didapati melakukan kesalahan kali kedua akan diambil tindakan seperti di perkara 2.1.2 dan dicadang untuk diambil tindakan tatatertib mengikut peruntukan Akta Universiti dan Kolej Universiti, 1971, Kaedah-kaedah Universiti Teknologi Malaysia (Tatatertib Pelajar-pelajar), 1999.

**SECTION A / *BAHAGIAN A***

**21 MARKS / *21 MARKAH***

**MULTIPLE CHOICE / *ANEKA PILIHAN***

**Choose the most appropriate answer. Write your answer in the table provided on page 6.**
***Pilih satu jawapan yang paling tepat. Tulis jawapan anda pada jadual di mukasurat 6.***

1. Choose the **incorrect** statement that describes an instruction cycle.
*Pilih penyataan yang **salah** yang menerangkan satu kitar arahan.*

   A. The execute cycle can happen before the fetch cycle.
   *Kitar laksana boleh berlaku sebelum kitar kutip.*

   B. The execute cycle can happen simultaneously with the fetch cycle in pipeline execution.
   *Kitar laksana boleh berlaku serentak dengan kitar kutip dalam perlaksanaan talian paip.*

   C. The content of the Program Counter (PC) or Instruction Pointer (IP) is updated before the next instruction is fetched.
   *Kandungan Pembilang Aturcara (PC) atau Penunjuk Arahan (IP) dikemaskini sebelum arahan dikutip.*

   D. Pre-fetch cycle occurs when the CPU needs to fetch operand from the memory.
   *Kitar pra-kutip berlaku apabila CPU perlu mengambil operan dari ingatan.*

2. Which control signal is issued by the control unit when fetching the instruction?
*Isyarat kawalan yang keluarkan oleh unit kawalan apabila mengutip arahan?*

   A. I/O Read / *Baca I/O*
   B. I/O Write / *Tulis I/O*
   C. Memory Read / *Baca Memori*
   D. Memory Write / *Tulis Memori*

3. I/O controller that has a second port to the main memory where the CPU does not need to perform the data transfer between the I/O and the memory is called _____.
*Pengawal I/O yang mempunyai port kedua ke ingatan di mana CPU tidak lagi melakukan operasi pemindahan data dari I/O ke ingatan dipanggil sebagai _____ .*

   A. SCSI port / *pot SCSI*
   B. USART / *USART*
   C. DMA controller / *pengawal DMA*
   D. Multiplexor channel / *saluran pemultipleks*

4. During the execution of a program, the decoded instruction is stored in _____
*Semasa perlaksanaan aturcara, arahan yang telah didekod disimpan dalam* _____ .

A. IR
B. PC
C. MAR
D. MDR

5. The address space of the IA-32 is _____ .
*Ruang alamat bagi IA-32 adalah* _____

A. $2^{16}$
B. $2^{32}$
C. $2^{64}$
D. $2^{8}$

6. When CPU perform subtraction on -7 and -5 the answer in 2's compliment form is _____.
*Apabila CPU melakukan penolakan ke atas -7 dan -5 jawapannya dalam pelengkap-2 ialah* _____ .

A. 0010
B. 1101
C. 1110
D. 1111

7. In DMA transfers, the required signals and addresses are given by the _____.
*Dalam pemindahan DMA, isyarat dan alamat yang diperlukan diberi oleh* _____.

A. Processor / *Pemproses*
B. Device drivers / *Pemacu peranti*
C. DMA controllers / *Pengawal DMA*
D. The program itself / *Aturcara itu sendiri*

8. Instruction like MOV or ADD is called _____.
*Arahan seperti MOV atau ADD dipanggil sebagai* _____ .

A. Op-Code / *Op-kod*
B. Operators / *Operator*
C. Command / *Perintah*
D. Assembler Directives / *Arahan Penghimpun*

9. Choose **illegal** instruction based on the following data definition:
*Pilih arahan yang **tidak sah** berdasarkan pentakrifan data berikut:*

```
.DATA
arrayW  WORD   1020h,3040h, 5060h
arrayD  DWORD  1,2,3,4
```

A. mov ax, arrayW+2
B. mov ax, arrayW [4]
C. mov ax, arrayD+4
D. mov eax, arrayD

10. Which of the following is **not** a legal instruction for the Intel Pentium CPU?
*Manakah di antara berikut **bukan** arahan yang sah bagi Intel Pentium CPU?*

A. add eax, ebx
B. add [eax], ebx
C. add [eax], [ebx]
D. add eax, [ebx]

11. Which instruction that will subtract val3 from EAX?
*Mana arahan yang akan kurangkan val3 dari EAX?*

A. sub val3, eax
B. sub eax, val3
C. subtract val3, eax
D. subtract eax, val3

12. Number of times the sequence of instruction below will loop before coming out of loop is
*Bilangan kali jujukan arahan di bawah akan dilakukan sebelum keluar daripada gelung*

```
      mov   eax, 0
A1:   INC   AL
      JNZ   A1
```

A. 00
B. 01
C. 255
D. 256

Given the following partial data segment which starts at address **000200H** and answer question 13 and 14.
*Diberi sebahagian segmen data yang bermula pada alamat **000200H** dan jawapan soalan 13 dan 14.*

```
.data
list    DWORD 1,2,6,24,120,720,5040,40320
x       DWORD  LENGTHOF list
y       DWORD SIZEOF list
```

13. What are the values of **x** and **y**?
*Apakah kandungan nilai **x** dan **y**?*

A. x = 8, y = 4
B. x = 8, y = 32
C. x = 32, y = 4
D. x = 32, y = 32

14. What is the address of **x**?
*Apakah alamat **x**?*

A. 000200H
B. 000208H
C. 000220H
D. 000232h

---

**ANSWER FOR SECTION A / *JAWAPAN UNTUK SEKSYEN A***
**21 MARKS / *MARKAH***

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | |

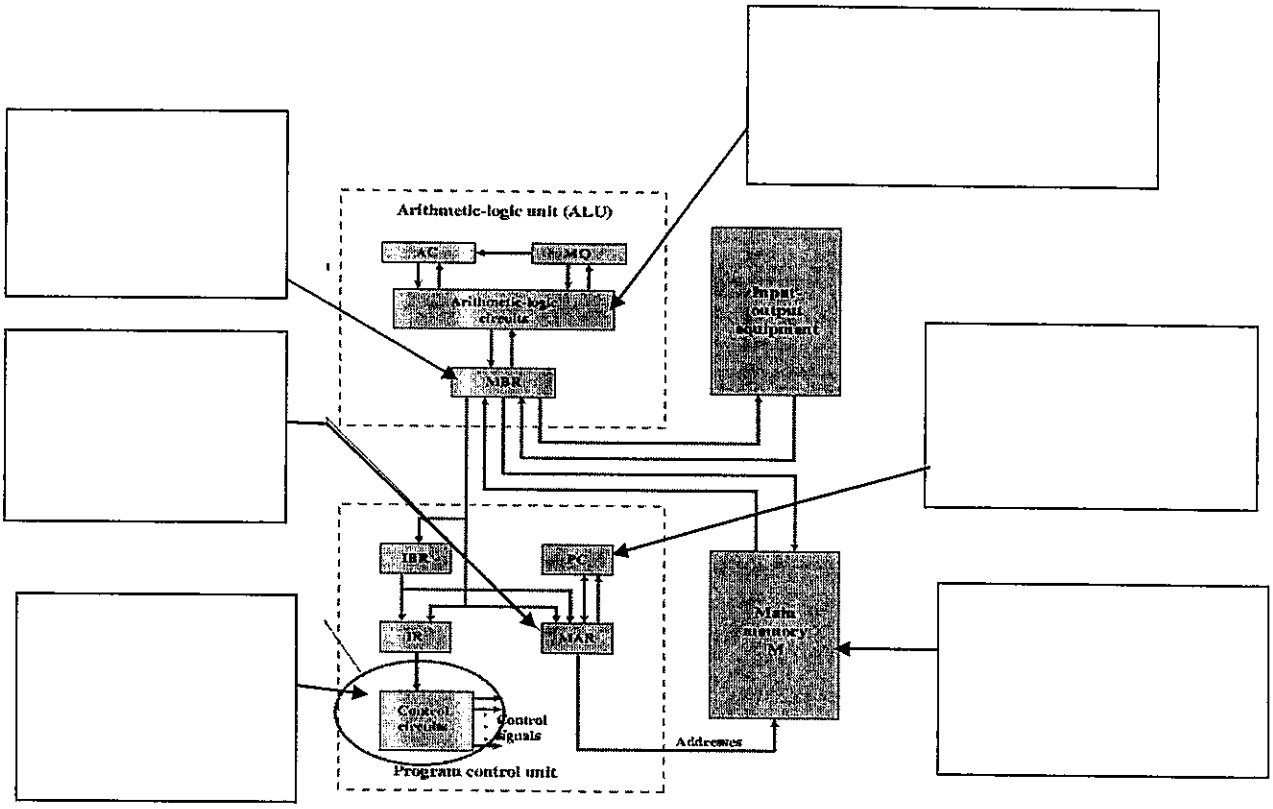
**SECTION B/ *BAHAGIAN B***

**79 marks / *79 markah***

**ANSWER ALL QUESTIONS. ANSWER IN THE PROVIDED SPACES IN THIS EXAM PAPER.**
***JAWAB SEMUA SOALAN. JAWAB PADA RUANG YANG DISEDIAKAN DALAM KERTAS SOALAN INI.***

Q1. Given the following block diagram of a computer architecture. Determine the function of each component indicated: **[6 M]**

*Diberi gambar rajah blok senibina komputer berikut. Tentukan fungsi setiap komponen yang dinyatakan:*



Q2. a) Explain the concept of a stored program computer. **[3 M]**

*Terangkan konsep komputer aturcara tersimpan.*

b) What is pipelining and how does it make the CPU executes faster? **[3 M]**

*Apakah itu talian paip dan bagaimana ia membuat CPU dapat melakukan perlaksanaan dengan lebih cepat?*

c) Explain how Intel implemented the concept of pipelining in the 8088/86 processor? **[3 M]**

*Jelaskan bagaimana Intel melaksanakan konsep talian paip di dalam pemproses 8088/86?*

Q3 a) Discuss what is cache memory. **[3 M]**
*Bincangkan apa itu ingatan cache.*

b) When the CPU needs data from the memory, the CPU will perform **Memory Read Cycle**. Write steps (in sequence) involved in a **Memory Read Cycle.** **[4 M]**
*Apabila CPU memerlukan data dari ingatan, CPU akan melakukan **Kitar Baca Ingatan.** Tuliskan langkah-langkah (mengikut turutan) yang terlibat dalam **Kitar Baca Ingatan.***

c) In order to execute a program, instruction must be transferred from memory along a bus to the CPU. If the bus has 16 data lines, at most two bytes (16 bit) can be transferred at a time. How many memory accesses would be needed in this case to transfer a 64 bit instruction from memory to the CPU. **[2 M]**
*Untuk melaksanakan satu arahan aturcara, arahan mesti dipindahkan dari ingatan utama ke CPU melalui bas. Jika sesuatu bas itu mempunyai 16 talian data, sebanyak dua bait (16 bit) data boleh dipindahkan pada satu masa. Berapa banyak capaian ingatan diperlukan dalam kes ini untuk memindahkan 64 bit arahan dari CPU.*

Q4. a) What is the main function of I/O interface module? **[2 M]**

*Apakah fungsi utama modul antaramuka I/O?*

b) Give two (2) differences between main memory and disk storage. **[4 M]**

*Berikan dua (2) perbezaan antara ingatan utama dan cakera storan.*

c) Why is DMA technique an improvement over programmed I/O technique? **[2 M]**

*Mengapa teknik DMA merupakan penambahbaikan ke atas teknik pemprograman I/O?*

d) When would DMA transfer be a poor choice? **[2 M]**

*Bilakah perpindahan DMA akan menjadi pilihan yang tidak bagus?*

Q5. a) Give the data declaration statements for the following:
*Berikan penyata deklarasi data bagi perkara-perkara berikut:*

i) Declare a **32-bit signed** integer variable and initialize it with the smallest possible negative decimal value. **[1 M]**
*Mengisytiharkan pembolehubah **32-bit integer bertanda** dan diberi nilai awalan dengan nilai perpuluhan negatif yang terkecil.*

ii) Declare a **string variable** containing "YEAR 2019" repeated 30 times, and terminated with the null character. **[1 M]**
*Mengisytiharkan **rentetan pembolehubah** mengandungi "YEAR 2019" yang diulang **30** kali, dan diakhiri dengan null.*

b) Trace the following program fragment and determine the content of the registers.
*Jejak fragmen aturcara berikut dan tentukan kandungan daftar.*

```
.DATA
Var1    dword  12345678h
Var2    dword  20003000h

.code
mov     eax, Var1
mov     ebx, Var2
xchg    ah,  al       ; al = _____ ah = _____
xchg    ax, bx        ; ax = _____ bx = _____
mul     123H          ; dx = _____ ax = _____
```

**[1 M]** **[1 M]** **[1 M]**

c) Write assembly language code to perform **AX = (-val2 + BX) - val4** **[4 M]**
*Tuliskan kod Bahasa himpunan untuk melakukan **AX = (-val2 + BX) - val4***

Q6. Show the content of the individual bytes allocated in memory (in hexadecimal) for the following data declarations. Assume a computer with 32-bit address bus, and that the physical address of **W** is **00808000h**. What are the addresses of **X**, **Y**, and **Z**? What is the total number of allocated bytes?
Note: ASCII Table attached in APPENDIX A **[10 M]**

*Tunjukkan kandungan setiap bait yang diperuntukkan dalam ingatan (dalam heksadesimal) untuk pengisytiharan data berikut. Andaikan komputer yang mempunyai 32-bit bas alamat, dan alamat fizikal **W** adalah **00808000h**. Apakah alamat **X**, **Y**, dan **Z**? Berapakah jumlah bait yang diperuntukkan?*
*Nota: Jadual ASCII di lampirkan pada APPENDIX A*

```
.DATA
W       BYTE    1,-1
X       SWORD   10FFh, -256
Y       DWORD   23456h
Z       BYTE    'DDWC 1223'
```

| address | content | address | content |
|---|---|---|---|
| 00808000H | | 0080800AH | |
| 00808001H | | 0080800BH | |
| 00808002H | | 0080800CH | |
| 00808003H | | 0080800DH | |
| 00808004H | | 0080800EH | |
| 00808005H | | 0080800FH | |
| 00808006H | | 00808010H | |
| 00808007H | | 00808011H | |
| 00808008H | | 00808012H | |
| 00808009H | | 00808013H | |

Address of X / *Alamat X*: _____

Address of Y / *Alamat Y*: _____

Address of Z / *Alamat Z*: _____

Number of bytes allocate / *Bilangan bait yang diperuntukkan*: _____

Q7. a) What is the value of **AX** and **CF** after the following program fragment is executed?
*Apakah nilai **AX** dan **CF** selepas keratan aturcara ini dilaksanakan?*

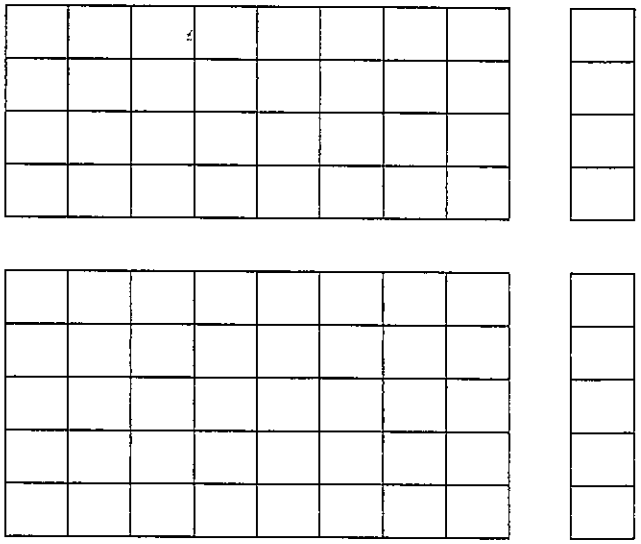
```
STC
XOR  EAX, EAX
MOV  AL, 0F
ADC  AL, 3Bh
SBB  AL, 37h
```

AX = _____ **[2 M]**

CF = _____ **[1 M]**

b) Let AX = 0000h, BX = 0000h and DX = 0000h. Indicate the content of register **AL**, **BL**, **DX** (all in hex value) and carry flag, **CF** after the execution of the following program fragment.
*Biarkan AX = 0000h, BX = 0000h dan DX = 0000h. Tunjukkan kandungan daftar **AL**, **BL** , **DX** (semua dalam nilai heksa) dan bendera bawa, **CF** selepas perlaksanaan keratan aturcara berikut.*

```
MOV  AX, 7C36h
MOV  BX, 9FA6h
MOV  DX, 0FF00h
SHR  AL, 3
SAR  BL, 4
XOR  DX, 00FFh
```



AL = _____ **[2 M]**

BL = _____ **[2 M]**

DX = _____ **[1 M]**

CF = _____ **[1 M]**

Q8. Trace the following program:
*Jejak aturcara berikut:*

```
TITLE Division                          (Divide.asm)
; This program divides  32-bit integers.
.686
.MODEL FLAT, STDCALL        .....................[1]
.STACK
INCLUDE Irvine32.inc
.CODE
main PROC
     xor edx, edx           ..................... [2]
     mov eax,00008003h
     mov ecx, 100h
     div cx
     call DumpRegs
     exit
main ENDP
END main
```

a) What is the use of **.MODEL** directive in instruction labeled [1] in the program above? **[1 M]**
*Apakah kegunaan direktif **.MODEL** pada arahan berlabel [1] dalam program di atas?*

_____

b) What is the output of instruction labeled [2] and the value of **OF** flag after the execution of instruction [2]? **[2 M]**
*Apa itu output arahan dilabel [2] dan nilai bendera **OF** selepas pelaksanaan arahan [2]?*

_____

c) After the execution of instruction **call DumpRegs** the following will be displayed. What will be the content of EAX and EDX? Answer by filling the blanks below. **[2 M]**
*Selepas perlaksanaan arahan **call DumpRegs** berikut akan paparkan. Apakah kandungan daftar EAX dan EDX? Jawab dengan mengisi ruang kosong di bawah:*

```
EAX=_____________   EBX=7FFDF000   ECX=00000100   EDX= _____________

ESI=00000000        EDI=00000000   EBP=0012FFF0   ESP=0012FFC4

EIP=00401024        EFL=00000206   CF=0  SF=0  ZF=0  OF=0
```

Q9. A scientist named Dr Aaron Kwak uses the following policy to normalize the temperature recorded in an experiment: "Normalization of temperatures is achieved by adding every temperature the difference between 99 and the highest temperature in the lab." If the following temperature is obtained: 81, 65, 77,82,73,55, 88, 78,51, 91, 86, and 76 (all values are in decimal), write a program that will:

i. Find the highest temperature. Put result in **HIGHEST.**

ii. Find the difference between 99 and the highest temperature recorded in the lab. Put result in **DIFF.**

iii. Normalize each temperature recorded and store it in array named **NORMALIZED.**

**Display** the result on the screen (that is the array NORMALIZED) using call DumpMem procedure. Your program **MUST** use **LOOP** and **JGE** instructions. Refer to Appendix B for Procedure calls.

Write your answer in the boxes provided below: **[12 M]**

*Seorang saintis bernama Dr Aaron Kwak menggunakan polisi berikut untuk menormalkan suhu yang dicatatkan dalam satu eksperimen: "Penomalan suhu dicapai dengan menambah setiap suhu dengan perbezaan di antara 99 dan suhu tertinggi di makmal." Jika suhu berikut diperolehi: 81, 65, 77,82,73,55, 88, 78,51, 91, 86 dan 76 (semua nilai adalah dalam decimal), tulis satu aturcara yang akan:*

*i. Mencari suhu tertinggi. Simpan hasil dalam **HIGHEST.***

*ii. Mencari perbezaan suhu di antara 99 dan suhu tertinggi yang direkodkan di makmal. Simpan hasil dalam **DIFF.***

*iii. Menormalkan setiap suhu yang direkod dan menyimpannya dalam tata susunan yang dinamakan **NORMALIZED.***

***Paparkan** hasil pada skrin (iaitu tata susunan **NORMALIZED**) dengan menggunakan prosedur DumpMem.untuk memapar kandungan ingatan. Program anda **MESTI** menggunakan arahan **LOOP** dan **JGE.** Rujuk Appendix B untuk memanggil Prosedur.*

*Tulis jawapan anda pada kotak yang disediakan di bawah:*

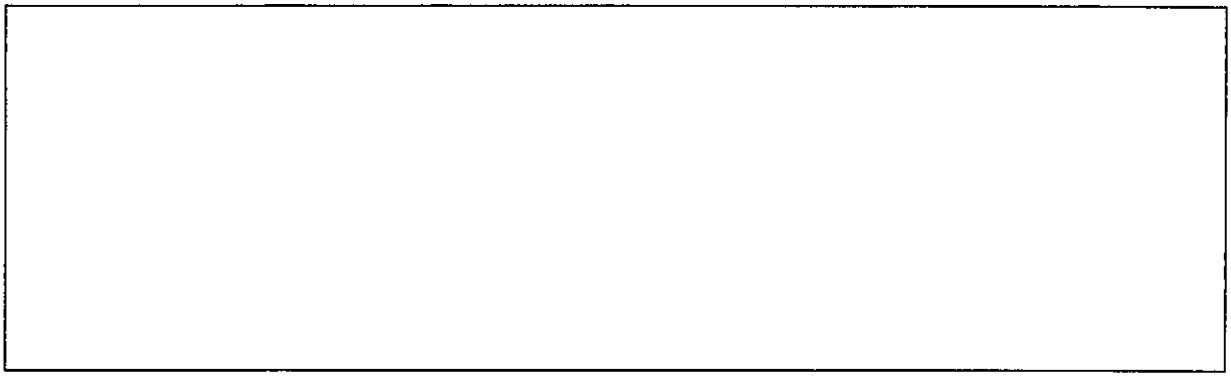
***ANSWER / JAWAPAN***

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

.data
TEMP        byte    81, 65, 77, 82, 73, 55, 88, 91, 86, 76
HIGHEST     byte    ?
NORMALIZED  byte    lengthof TEMP dup(?)
DIFF        byte    ?
```

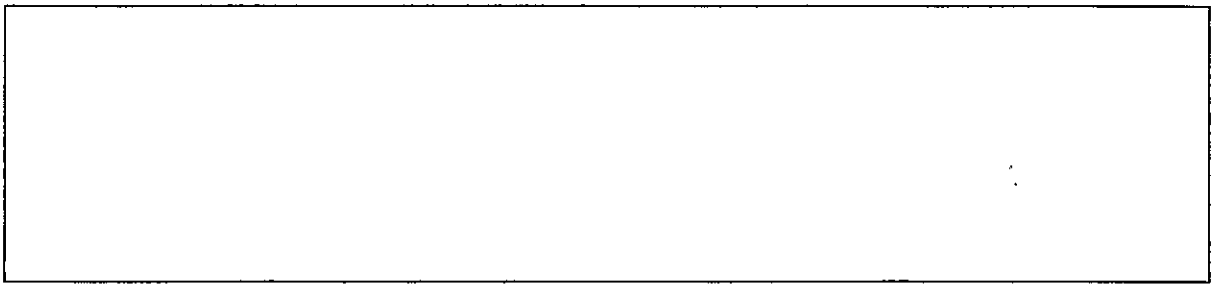
```
.code

main proc
 ; to find the highest value in array named TEMP and store it in HIGHEST
```

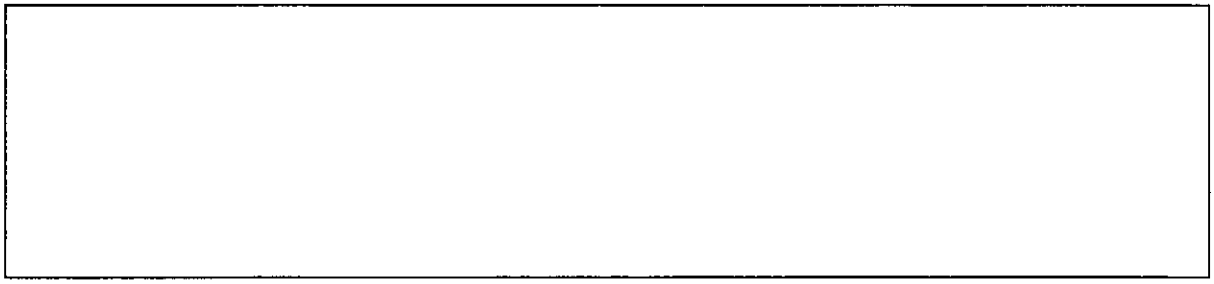


```
 ; to find the difference between highest value and 99 that is to find DIFF
```

```
 ; to normalize the temperature and store result in NORMALIZED
```



```
 ; displaying the content of NORMALIZED
```



```
exit
main endp
end main
```

## APPENDIX A

ASCII TABLE / *JADUAL ASCII*

| Dec | Hex | Char | Dec | Hex | Char | Dec | Hex | Char | Dec | Hex | Char |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 00 | Null | 32 | 20 | Space | 64 | 40 | @ | 96 | 60 | ` |
| 1 | 01 | Start of heading | 33 | 21 | ! | 65 | 41 | A | 97 | 61 | a |
| 2 | 02 | Start of text | 34 | 22 | " | 66 | 42 | B | 98 | 62 | b |
| 3 | 03 | End of text | 35 | 23 | # | 67 | 43 | C | 99 | 63 | c |
| 4 | 04 | End of transmit | 36 | 24 | $ | 68 | 44 | D | 100 | 64 | d |
| 5 | 05 | Enquiry | 37 | 25 | % | 69 | 45 | E | 101 | 65 | e |
| 6 | 06 | Acknowledge | 38 | 26 | & | 70 | 46 | F | 102 | 66 | f |
| 7 | 07 | Audible bell | 39 | 27 | ' | 71 | 47 | G | 103 | 67 | g |
| 8 | 08 | Backspace | 40 | 28 | ( | 72 | 48 | H | 104 | 68 | h |
| 9 | 09 | Horizontal tab | 41 | 29 | ) | 73 | 49 | I | 105 | 69 | i |
| 10 | 0A | Line feed | 42 | 2A | * | 74 | 4A | J | 106 | 6A | j |
| 11 | 0B | Vertical tab | 43 | 2B | + | 75 | 4B | K | 107 | 6B | k |
| 12 | 0C | Form feed | 44 | 2C | , | 76 | 4C | L | 108 | 6C | l |
| 13 | 0D | Carriage return | 45 | 2D | - | 77 | 4D | M | 109 | 6D | m |
| 14 | 0E | Shift out | 46 | 2E | . | 78 | 4E | N | 110 | 6E | n |
| 15 | 0F | Shift in | 47 | 2F | / | 79 | 4F | O | 111 | 6F | o |
| 16 | 10 | Data link escape | 48 | 30 | 0 | 80 | 50 | P | 112 | 70 | p |
| 17 | 11 | Device control 1 | 49 | 31 | 1 | 81 | 51 | Q | 113 | 71 | q |
| 18 | 12 | Device control 2 | 50 | 32 | 2 | 82 | 52 | R | 114 | 72 | r |
| 19 | 13 | Device control 3 | 51 | 33 | 3 | 83 | 53 | S | 115 | 73 | s |
| 20 | 14 | Device control 4 | 52 | 34 | 4 | 84 | 54 | T | 116 | 74 | t |
| 21 | 15 | Neg. acknowledge | 53 | 35 | 5 | 85 | 55 | U | 117 | 75 | u |
| 22 | 16 | Synchronous idle | 54 | 36 | 6 | 86 | 56 | V | 118 | 76 | v |
| 23 | 17 | End trans. block | 55 | 37 | 7 | 87 | 57 | W | 119 | 77 | w |
| 24 | 18 | Cancel | 56 | 38 | 8 | 88 | 58 | X | 120 | 78 | x |
| 25 | 19 | End of medium | 57 | 39 | 9 | 89 | 59 | Y | 121 | 79 | y |
| 26 | 1A | Substitution | 58 | 3A | : | 90 | 5A | Z | 122 | 7A | z |
| 27 | 1B | Escape | 59 | 3B | ; | 91 | 5B | [ | 123 | 7B | { |
| 28 | 1C | File separator | 60 | 3C | < | 92 | 5C | \ | 124 | 7C | \| |
| 29 | 1D | Group separator | 61 | 3D | = | 93 | 5D | ] | 125 | 7D | } |
| 30 | 1E | Record separator | 62 | 3E | > | 94 | 5E | ^ | 126 | 7E | ~ |
| 31 | 1F | Unit separator | 63 | 3F | ? | 95 | 5F | _ | 127 | 7F | □ |

## APPENDIX B

## Link library Irvine32.lib

### 1. Dumping Registers and Memory

| Procedure | Description |
|---|---|
| DumpRegs | Writes EAX, EBX, ECX, and EDX on first line in hexadecimal<br>Writes ESI, EDI, EBP, and ESP on second line in hexadecimal<br>Writes EIP, EFLAGS, CF, SF, ZF, and OF on third line |
| DumpMem | Writes a range of memory to standard output in hexadecimal<br>ESI = starting address<br>ECX = number of elements to write<br>EBX = element size (1, 2, or 4) |

### 2. Input procedures: ReadInt, ReadChar, ReadString,Dec

| Procedure | Description |
|---|---|
| ReadChar | Reads a char from keyboard and returns it in the AL register.<br>The character is NOT echoed on the screen. |
| ReadHex | Reads a 32-bit hex integer and returns it in the EAX register.<br>Reading stops when the user presses the [Enter] key.<br>No leading spaces. No error checking is performed. |
| ReadInt | Reads a 32-bit signed integer and returns it in EAX.<br>Leading spaces are ignored. Optional + or – is allowed.<br>Error checking is performed (error message) for invalid input. |
| ReadDec | Reads a 32-bit unsigned integer and returns it in EAX. |
| ReadString | Reads a string of characters from keyboard.<br>Additional null-character is inserted at the end of the string.<br>EDX = address of array where input characters are stored.<br>ECX = maximum characters to be read + 1 (for null byte)<br>Return EAX = count of non-null characters read. |

3. **Output procedures: Clrscr, WriteInt, WriteHex, WriteString,WriteHex, WriteInt, WriteDec, WriteBin**

| Procedure | Description |
|---|---|
| Clrscr | Clears screen, locates cursor at upper left corner. |
| Crlf | Writes end of line sequence (CR,LF) to standard output. |
| WriteChar | Writes character in register AL to standard output. |
| WriteString | Writes a null-terminated string to standard output.<br>String address should be passed in register EDX. |
| WriteHex | Writes EAX in hexadecimal format to standard output. |
| WriteInt | Writes EAX in signed decimal format to standard output. |
| WriteDec | Writes EAX in unsigned decimal format to standard output. |
| WriteBin | Writes EAX in binary format to standard output. |

END OF QUESTIONS / *SOALAN TAMAT*

**Mukasurat ini sengaja dibiarkan kosong**

***[ This page is purposely left blank ]***