

Comenzado el	Thursday, 26 de May de 2022, 08:40
Estado	Finalizado
Finalizado en	Thursday, 26 de May de 2022, 10:12
Tiempo empleado	1 hora 32 minutos
Calificación	16,11 de 20,00 (81%)

Pregunta **1**

Finalizado

Se puntúa 5,00
sobre 6,00

Cree una subrutina reusable que reciba una cadena de caracteres por referencia a través de la pila y reemplace con '&' los caracteres '%' que encuentre. Se debe almacenar la cantidad de reemplazos realizados en un parámetro por referencia enviado vía registro.

ORG 3000H

SUBROUTINA: PUSH BX

MOV BX, SP

ADD BX, 4

PUSH AX

PUSH CX

PUSH DX

MOV AX, BX ; DIREC. TABLA EN PILA

MOV BX, [BX] ; DIREC. TABLA EN MEMORIA

REPEAT: MOV CH, [BX] ; VALOR DE LA POS X DE LA TABLA

CMP CH, '%'

JZ REEMPLAZAR

JMP ELSE

REEMPLAZAR: MOV [BX] , '&'

INC CL

ELSE: INC BX

CMP BX, OFFSET FIN_CADENA

JNZ REPEAT

MOV BX, DX ; 1

MOV [BX], CL ; 2

; en 1 y 2 almaceno cantidad de reemplazos en parámetro por referencia enviado vía registro

POP DX

POP CX

POP AX

POP BX

RET

Comentario:

Justina te mando las correcciones:

- MOV AX, BX ; DIREC. TABLA EN PILA
MOV BX, [BX] ; DIREC. TABLA EN MEMORIA. En estas líneas tenías que volver a mover lo que guardaste en AX a BX para poder hacer la indirección ó directamente usar la línea 2, sin pasar por AX.
- JZ REEMPLAZAR
JMP ELSE. Este caso, también, funciona pero esta "raro", podrías haber reacomodado el código para que no tengas que usar dos jumps seguidos.

Pregunta **2**
Parcialmente
correcta

Se puntúa 2,73
sobre 3,00

Dado el siguiente programa, indique los modos de direccionamiento utilizados en las instrucciones indicadas:

ORG 1000H
TABLA DW 1,1,1,1,1,2
CANT DW 2

ORG 3000H
ROTAR: PUSH CX
PUSH DX

MOV BX, SP ; Directo de registro ✓

ADD BX,10 ; Inmediato ✓

MOV BX,[BX]
MOV AX,[BX] ; Indirecto con registro ✓

REPE: CMP AX, 0
JZ FR

ADD AX,AX
MOV [BX],AX
ADD BX,2

MOV AX,[BX] ; Indirecto con registro ✓

JMP REPE ; Directo de memoria ✗ INMEDIATO? porq le dice anda a tal direc (x ej 1002h)

FR: POP DX
POP CX
RET

ORG 2000H
MOV AX,1000H ; Inmediato ✓
ADD AX, CANT ; Directo de memoria ✓
PUSH AX
MOV CX,OFFSET CANT ; Inmediato ✓

MOV BX,CX
MOV BX,[BX] ; Indirecto con registro ✓

LOOP: CMP BX,0
JZ FIN

PUSH BX
CALL ROTAR
POP BX

DEC BX ; Directo de registro ✓

JMP LOOP
POP AX ; Directo de registro ✓

FIN:HLT
END

Pregunta **3**
Parcialmente
correcta
Se puntúa 0,80
sobre 1,00

Indique para cada característica si la misma se corresponde con la localidad espacial o la localidad temporal

Subrutinas (Procedimientos o Funciones)	Temporal	✓
Tendencia de los programadores a hacer próximas las variables relacionadas	Espacial	✓
Ejecución secuencial del código	Temporal	✗
Acceso a estructuras tipo matriz	Espacial	✓
Uso de while	Temporal	✓

Respuesta parcialmente correcta.
Ha seleccionado correctamente 4.
La respuesta correcta es: Subrutinas (Procedimientos o Funciones) → Temporal, Tendencia de los programadores a hacer próximas las variables relacionadas → Espacial, Ejecución secuencial del código → Espacial, Acceso a estructuras tipo matriz → Espacial, Uso de while → Temporal

Pregunta **4**
Parcialmente
correcta
Se puntúa 0,83
sobre 1,00

Complete cada palabra en los párrafos según lo visto en el curso.

Los lenguajes de programación de bajo nivel son ✓ de la CPU, están hechos a medida del ✗ de la misma y por lo tanto aprovechan al máximo sus características.

El programa escrito en lenguaje ensamblador es de ✓ lectura ya que su estructura se acerca al lenguaje máquina.

El lenguaje ensamblador es ✓ portable, es decir, un código escrito para un ✓ , puede requerir modificación, para poder ser usado en otra máquina distinta del mismo fabricante.

Los programas creados por un programador experto en lenguaje ensamblador son generalmente mucho más ✓ y consumen menos memoria de datos y memoria de programa que el programa equivalente proveniente de un lenguaje de alto nivel.

Respuesta parcialmente correcta.
Ha seleccionado correctamente 5.
La respuesta correcta es:
Complete cada palabra en los párrafos según lo visto en el curso.
Los lenguajes de programación de bajo nivel son [dependientes] de la CPU, están hechos a medida del [hardware] de la misma y por lo tanto aprovechan al máximo sus características.
El programa escrito en lenguaje ensamblador es de [difícil] lectura ya que su estructura se acerca al lenguaje máquina.
El lenguaje ensamblador es [difícilmente] portable, es decir, un código escrito para un [microcontrolador], puede requerir modificación, para poder ser usado en otra máquina distinta del mismo fabricante.
Los programas creados por un programador experto en lenguaje ensamblador son generalmente mucho más [rápidos] y consumen menos memoria de datos y memoria de programa que el programa equivalente proveniente de un lenguaje de alto nivel.

Pregunta **5**

Correcta

Se puntúa 0,50
sobre 0,50

En la codificación BIG ENDIAN el byte más significativo ocupa la dirección de memoria con valor numérico más bajo

Seleccione una:

- ☒ Verdadero ✓
- ☐ Falso

La respuesta correcta es 'Verdadero'

Pregunta **6**

Incorrecta

Se puntúa 0,00
sobre 0,50

La memoria de disco es una memoria costosa y lenta.

Seleccione una:

- ☒ Verdadero ✗
- ☐ Falso

La respuesta correcta es 'Falso'

Pregunta **7**

Correcta

Se puntúa 0,50
sobre 0,50

Los registros AX, BX, CX y DX son de uso general, 16 bits de longitud y se pueden dividir en 2 partes de 8 bits cada uno.

Seleccione una:

- ☒ Verdadero ✓
- ☐ Falso

La respuesta correcta es 'Verdadero'

Pregunta **8**

Correcta

Se puntúa 0,50
sobre 0,50

La dirección de la instrucción que se debe captar durante la ejecución de un programa se encuentra en el stack pointer (SP).

Seleccione una:

- ☐ Verdadero
- ☒ Falso ✓

La respuesta correcta es 'Falso'

Pregunta **9**

Finalizado

Se puntúa 2,00
sobre 2,00

Describe la memoria que corresponde a la pila colocando las direcciones de memoria y los valores guardados durante toda la ejecución del programa.

```
ORG 1000H; Datos
TABLA DW 1,2,3,4,5,6
VALOR DW 8

ORG 3000H; Subrutina
SUMA: PUSH CX
MOV BX, SP
ADD BX, 6
MOV CX, [BX]
ADD BX, 2
POP CX      ; (1)
DEC DX
RET

ORG 2000H; Programa
MOV AX, OFFSET TABLA
MOV BX, VALOR
MOV DL, OFFSET VALOR - OFFSET TABLA
CMP DL, 0
JZ FIN
PUSH AX
PUSH BX
CALL SUMA
POP BX
POP AX
FIN: HLT
END
```

- Comienza el SP en 4000H
- SP -> 3FFEh ; [SP] -> 1000H (OFFSET TABLA)
- SP -> 3FFCh ; [SP] -> 8 (VALOR)
- CALL SUMA
- SP -> 3FFAh ; [SP] -> DIR. DE RETORNO
- PUSH CX: SP -> 3FF8H ; [SP] -> VALOR DE CX
- POP CX: SP -> 3FFAh ; [SP] -> DIR. DE RETORNO
- RET
- SP -> 3FFCh ; [SP] -> 8 (VALOR)
- POP BX: SP -> 3FFEh ; [SP] -> 1000H (OFFSET TABLA)
- POP AX: SP -> 4000H (END)

Comentario:

Este ejercicio está muy bien!

Pregunta **10**

Parcialmente correcta

Se puntúa 2,25 sobre 3,00

Completar el programa para que la rutina Suma reciba un parámetro A por valor vía pila y B sea enviado por valor vía registro:

ORG 1000H ;Datos

A DW OFFEH

B DB 15h

ORG 3000H ; Subrutina

Suma: ... ; otro procesamiento

FinSuma: RET ; vuelvo al programa

ORG 2000H ;Programa

MOV AX, A ✓

MOV BX, OFFSET B ✓

PUSH AX

MOV BX ✗ , [BX]

CALL Suma

POP ✓ AX

... ; continua el programa

HLT

END

Pregunta **11**

Correcta

Se puntúa 1,00 sobre 1,00

¿Qué estructuras de control existen en el siguiente programa?

```
ORG 1000H
TABLA DB 5, 50, 15, 20, 2

ORG 2000H
MOV BX, OFFSET TABLA
MOV DL, 15
MOV CL, 0
ETIQUETA: ADD CL,[BX]
INC CL
INC BX
CMP DL,[BX]
JNZ ETIQUETA
FIN: HLT
END
```

Seleccione una o más de una:

- ☐ a. while
- ☒ b. repeat-until ✓
- ☐ c. if-then
- ☐ d. if-then-else
- ☐ e. for

Respuesta correcta

La respuesta correcta es: repeat-until

Pregunta **12**

Incorrecta

Se puntúa 0,00 sobre 1,00

¿Qué estructuras de control existen en el siguiente programa?

```
ORG 1000H
DATO DB 0H

ORG 2000H
CMP AL,DATO
JZ ETIQUETA
JMP FIN
ETIQUETA: MOV BL, 1
INC CL
FIN: HLT
END
```

Seleccione una o más de una:

- ☐ a. if-then
- ☐ b. repeat-until
- ☒ c. if-then-else ✖
- ☐ d. while
- ☐ e. for

Respuesta incorrecta.

La respuesta correcta es: if-then