Ejercicios BM2 05nov19

Procedimientos, pasaje de parámetros por stack.

En los procedimientos (subrutinas, funciones o métodos) debe escribir como comentarios, como siempre: la funcionalidad del procedimiento, los parámetros recibidos, los resultados devueltos y algún requerimiento.

Recuerde que al invocar un procedimiento, los parámetros efectivos son pasados por el stack, como se hace en los lenguajes de alto nivel. El resultado, si es que lo hay explícitamente, deberá ser dejado en el stack, por el procedimiento, y recogido del stack por el programa invocador, p.e. main.

Lea siempre, todo el ejercicio completo antes de empezar a resolverlo.

Los datos y resultados deben ser precedidos por un texto adecuado.

En los siguientes ejercicios deben quedar bien claras las estructuras algorítmicas empleadas (ahora con las macro directivas).

1. Se tienen las siguientes dos listas (arreglos paralelos) definidos en el programa con la directiva SDWORD dentro de .DATA. En este ejemplo, cada uno de los dos arreglos tiene 7 elementos, pero piense el ejercicio como si fueran “n” elementos.

|  |
| --- |
| **arr1 arr2**  11 209  10 -131  12 -96  14 160  13 -221  10 85  12 -49 |

Calcule la sumatoria de los productos de cada renglón (11\*209+10\*(-131)+. . .).

Se requiere, de un procedimiento *SumaMulti(arr1, arr2, tamArr)*, que vaya imprimiendo el cálculo de cada nuevo término, mientras va acumulando el total, devolviendo antes de RET, como resultado la sumatoria de todos los términos.

Al final el programa principal imprimirá tanto el resultado de la sumatoria como el texto ADIOS.

1. Calcule la longitud del *texto* “El coronel no tiene quien le escriba”, que se encuentra en el segmento .DATA, así como la cantidad de caracteres “e”, “n”, “u” y “r” que contiene dicha cadena, en un programa ensamblador.

Se cuenta con las siguientes declaraciones en ensamblador.

|  |
| --- |
| .DATA  texto BYTE “El coronel no tiene ”  BYTE “quien le escriba.”  BYTE 0  conul BYTE “enur” ; caracteres a contabilizar  acont DWORD 4 DUP (?) ; total de cada carácter |

El ejercicio deberá pensarlo como si el total de caracteres del texto fueran “m”.

Para este ejercicio usted desarrollará, además del procedimiento *main*, cuatro procedimientos más:

1. El primero, que calcule el total de caracteres que tiene la cadena, excluyendo el carácter nulo. La cadena se encuentra inicialmente en *texto* del procedimiento *main*.
2. El segundo, que imprima toda la cadena y su longitud, en un mismo renglón, acompañado de textos adecuados. Ver figura 1.
3. El tercero, que cuente el total de caracteres “e”, “n”, “u” y “r” que tiene la cadena; dichos caracteres se encuentran en el arreglo de bytes *conul* de *main*. Respectivamente, en el orden que se encuentran “e”, “n”, “u” y “r”, deje las cuentas en los elementos del arreglo *acont*.
4. Finalmente, el cuarto procedimiento deberá imprimir el total, de cada una de las cuatro letras en forma de tabla.

Si el programa fuera escrito en lenguaje de alto nivel luciría como sigue:

|  |
| --- |
| main {  String texto = “El coro . . .”;  byte cenur[] = . . .  int acont[];  int lon, f;  lon = totcar(texto);  imprimetyl(texto, lon);  ctaenur(texto, lon, conul, acont);  imprimecta(conul, acont);  System.out.println(“Adios”);  }  int totcar(String cad) {  int tam;  ; código del programa  return tam;  }  void imprimetyl(String txt, int tam) {  ;código del programa  }  void ctaenur(String txt, int tam,  byte car[], int ac[]) {  ; código del programa  }  void imprimecta(byte car[], int ac[]) {  ; código del programa  } |

Continua en la siguiente página.

El programa lo deberá desarrollar en lenguaje ensamblador.

Imprima ADIOS al final.

El pasaje de parámetros deberá hacerlo al estilo de los lenguajes de alto nivel, considerando que en lugar de un arreglo debe pasar su dirección de memoria correspondiente (OFFSET).

El despliegue tendría el siguiente aspecto.

|  |
| --- |
| Texto: El coronel no tiene quien le escriba. Longitud: 37 caracteres  Carácter cantidad  e 6  n 4  u 1  r 2 |