

2do Parcial Principios de Programación

12/07/2011

Ejercicio 1) (25 puntos)

Dado un arreglo con tope de enteros definido como

```
#define cant_elem 100

struct arreglo_con_tope{
    int arreglo[cant_elem];
    int cantidad;
};
```

Donde cantidad indica la cantidad de elementos en el arreglo.

Se pide implementar las siguientes funciones:

a)

```
void inicializarArreglo(struct arreglo_con_tope & numeros)
```

Inicializa el arreglo con tope parámetro (numeros), de forma tal que represente que está vacío.

b)

```
void cargarArreglo(struct arreglo_con_tope & numeros)
```

Lee una secuencia de números enteros positivos y los carga en el arreglo con tope parámetro (numeros). El fin de la secuencia está indicado por el valor -1.

Se puede asumir que la secuencia no tiene más de 100 números, no es vacía y que el usuario solo va a ingresar enteros positivos a excepción del -1 para indicar el fin de la secuencia. También se puede asumir que previo a la invocación de esta función se llamo a la función inicializarArreglo con el arreglo numeros.

c)

```
int sumaValoresArreglo(struct arreglo_con_tope numeros)
```

Calcula y devuelve la suma de los valores contenidos en el arreglo con tope parámetro (numeros).

Se puede asumir que previó a la invocación de esta función se llamo a la función cargarArreglo con el arreglo numeros.

d)

Defina la función main que realice lo siguiente invocando donde corresponda a las funciones definidas en las partes a, b y c.

Debe inicializar un arreglo con tope representando que está vacío.

Debe pedir al usuario que ingrese una secuencia de números enteros positivos finalizando la misma con -1.

Debe guardar la secuencia de números ingresados en el arreglo con tope previamente inicializado.

Debe sumar los valores cargados en el arreglo con tope.

Debe mostrar en pantalla el resultado de la suma.

Ejercicio 2) (20 puntos)

Considerando el problema a resolver en el Obligatorio 2, se pide implementar las siguientes funciones:

bool tieneParentesis(char expre[])

Recibe una expresión matemática (expre) y devuelve true si la misma tiene algún paréntesis (puede ser de apertura o de cierre) y false en caso contrario.

bool parentesisOk(char expre[])

Recibe una expresión matemática (expre) y devuelve true si esta bien formada con respecto a los paréntesis y false en caso contrario. Podemos asumir que expre tiene al menos un paréntesis.

Ejercicio 3) (15 puntos)

Indicar cual es la salida en pantalla al ejecutar el siguiente fragmento de programa:

```
int i=0;

void f(){
    printf("imprimo i en f %d\n",i);
}

void g(int & i){
    printf("imprimo i en g %d\n",i);
    i=20;
}

void h(){
    static int i=5;
    printf("imprimo i en h %d\n",i);
}

main() {
    int i=10;
    printf("imprimo i en main (1) %d\n",i);
    f();
    g(i);
    h();
    f();
    printf("imprimo i en main (2) %d\n",i);
    h();
    system("PAUSE");
}
```