

2361 - PROGRAMACION II - MAÑANA A - 2Q

 > [Evaluación](#) > Prueba 1 - Temas 1 al 7

1/1

Comenzado el	viernes, 8 de marzo de 2024, 09:15
Estado	Finalizado
Finalizado en	viernes, 8 de marzo de 2024, 10:15
Tiempo empleado	1 hora
Calificación	5,60 de 10,00 (56%)

Pregunta 1

Finalizado

Se puntúa 2,00 sobre 2,00

Escribe una función que reciba un entero n y muestre por pantalla un triángulo que comenzará con un solo carácter '*' y cuya base tendrá n caracteres, de manera que cada línea del triángulo tenga un carácter más hasta llegar a la base. Los caracteres de las líneas tendrán que estar alineados con el carácter más a la derecha de la última fila.

Ejemplo de triángulo para $n = 5$:

```
*  
**  
***  
****  
*****
```

```
void triangulo(int n) {  
    int base = (n+1)/2;  
    for (int i = 0; i < n; ++i) {  
        for (int j = n-i-1; j >= 0; --j) {  
            printf(" ");  
        }  
        for (int j = 0; j < i+1; ++j) {  
            printf("*");  
        }  
        printf("\n");  
    }  
}
```

Comentario:

Pregunta 2

Finalizado

Se puntúa 1,60 sobre 2,00

Escribe una función que reciba dos valores a y b , de manera que genere una serie de a valores **que se creará en un array local a la función** donde el primer valor siempre será 1, y el segundo valor y siguientes serán el valor anterior más b .

Ejemplo: $a = 5$, $b = 3$ genera la serie 1 4 7 10 13

La función debe generar la serie, mostrarla por pantalla y mostrar debajo la serie invertida.

Ejemplo: para $a = 5$, $b = 3$, primero la serie y después la serie invertida.

```
1 4 7 10 13
13 10 7 4 1
```

Los valores permitidos para los parámetros son $1 < a < 10$ y $-5 < b < 5$. Si el valor que recibe la función para los parámetros no se encuentra en estos rangos se deberá mostrar un mensaje de error en pantalla y no se calculará la serie.

```
void valores(int a, int b) {
    int *nums;
    nums = calloc(a, sizeof(int));
    nums[0] = 1;
    for (int i = 1; i < a; ++i) {
        nums[i] = nums[i-1] + b;
    }

    printf("Serie normal:\n");
    for (int i = 0; i < a; ++i) {
        printf(" %i ", nums[i]);
    }

    printf("\n");

    printf("Serie invertida:\n");
    for (int i = a-1; i >= 0; --i) {
        printf(" %i ", nums[i]);
    }
}
```

Comentario:

Falta comprobación de valores de parámetros.

Como has usado calloc, tendrías que haber hecho free!!

Pregunta 3

Finalizado

Se puntúa 0,00 sobre 3,00

Crea una función que reciba como argumento un número entero n , que será el número de palabras con el que tiene que trabajar. La función pedirá al usuario las n palabras y, una vez introducidas, contará el número de vocales que hay en cada palabra, mostrando ese número después de la palabra introducida. La función deberá devolver como valor de retorno el número total de vocales.

Ejemplo de interacción con el usuario para $n = 3$:

Introduzca 3 palabras, una por línea:

la

sopa

quema

El número de vocales es 6, que se distribuyen así:

la : 1

sopa : 2

quema : 3

Las palabras no podrán tener una longitud mayor de 20 caracteres y se escribirán siempre en minúsculas y sin acentos. Puedes intentar resolver el problema para una sola palabra y después modificar la función para que gestione n palabras.

Comentario:
No entregado.

Pregunta 4

Finalizado

Se puntúa 0,50 sobre 1,00

Escribe una función equivalente a la siguiente, pero SIN utilizar la función *realloc*.

```
int *duplicaArray(int tamanoActual, int *p) {  
  
    p = (int *) realloc(p, 2 * tamanoActual * sizeof(int));  
    if (p == NULL) {  
        printf("No se pudo reservar memoria.\n");  
    }  
    return p;  
}
```

```
int *duplicaArray(int tamano, int *p) {  
    p = (int *) calloc(tamano, sizeof(int));  
    if (p == NULL) {  
        printf("No hay memoria");  
        return NULL;  
    }  
    for (int i = 0; i < tamano; ++i) {  
        p[i] = i+1;  
    }  
    int *newp = (int *) calloc(tamano*2, sizeof(int));  
    if (newp == NULL) {  
        printf("No hay memoria");  
        return NULL;  
    }  
    for (int i = 0; i < tamano; ++i) {  
        newp[i] = p[i];  
    }  
    int i = 0;  
    /*  
    while (p[i] != tamano) {  
        newp[i+5] = p[i];  
        i += 1;  
    }  
    */  
    do {  
        newp[i+5] = p[i];  
        i += 1;  
    } while (p[i] != tamano);  
  
    return newp;  
}
```

Comentario:

En el inicio de tu función desvinculas la memoria ya reservada en p, y creas un nuevo vector, que rellenas innecesariamente.

Después haces bien la reserva de nueva memoria y la copia (esto es correcto), devolviendo el nuevo puntero.

Sin embargo, rellenas en newp valores a partir del 5 ¿¿??

Los errores penalizan, además de que no hay free del viejo puntero.

Pregunta 5

Finalizado

Se puntúa 1,50 sobre 2,00

Suponiendo que las funciones escritas en los ejercicios 1 al 3 estuvieran codificadas en un fichero se pide codificar la función *main* que invocaría a esas funciones para mostrar su funcionamiento según la elección del usuario.

Para ello, el programa recibirá dos argumentos a través de su llamada desde el terminal: el nombre de la función a la que invocar (*triangulo*, *serie-suma* o *cuenta-vocales*) y un número, que será el argumento a usar en la llamada a la función.

El código deberá procesar esos argumentos de entrada e invocar a la función correspondiente solamente en el caso en que se reciban los dos parámetros. En el caso de "*serie-suma*" será necesario comprobar que se recibe un parámetro adicional, que se usará en la llamada a la función.

```
int main(int argc, char *argv[]) {
    printf("Parametros: %i", argc);
    for (int i = 0; i < argc; ++i) {
        printf("Argumento %i es: %s", i, argv[i]);
    }
    int param3 = atoi(argv[3]);
    int param2 = atoi(argv[2]);
    char *func = argv[1];
    if (strcmp(func, "triangulo") == 0) {
        triangulo(param2);
    }
    if (strcmp(func, "serie-suma") == 0) {
        valores(param2, param3);
    }
    if (strcmp(func, "cuenta-vocales") == 0) {
        palabras(param2);
    }
    return 0;
}
```

Comentario:

Da violación de segmento si no se meten tres parámetros porque no hay comprobación de número de parámetros.

Falta incluir librería string.h.

Actividad previa

Ejercicio: manejo de repositorios (auto corregible)

Ir a...

Siguiente actividad

---- [SOLUCIÓN Prueba 1](#)