

PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

Trabajo Práctico N° 8

TEMA: Mecanismos de Pasaje de Parámetros - Asignación de memoria dinámica



*Para los desplazamientos en los arreglos, preferentemente notación de punteros.
Siempre que sea posible declare los arreglos de forma dinámica.*

Problema 1: Cadenas apuntadas

Implemente un programa que realice las siguientes operaciones en una cadena:

- a. **Función Buscar:** operación que le permite determinar si la cadena contiene un determinado carácter. La función debe devolver un valor de verdad y además, la cantidad de veces que el carácter aparece. Use el siguiente prototipo:

int buscar(char *cad, char c, int *cant);

- b. **Función Reemplazar:** permite reemplazar un carácter en una determinada posición de la cadena. Use el siguiente prototipo:

void reemplazar (char *cad, int pos, char c);

- c. **Función Insertar:** genera una nueva cadena insertando un carácter determinado después de la aparición del mismo carácter en la cadena original.
Esta nueva cadena debe generarse de forma dinámica. Reutilice la función buscar de ser necesario.

Use el siguiente prototipo: **void insertar(char *cad, char *nueva, char c);**

Ejemplo:

Frase original: "aprender programacion es muy facil"

Carácter a insertar: **a**

Nueva frase: "aaprender prograamaacion es muy faacil"

Problema 2: Cadena procesada

Escribir un **único** módulo que dada una frase, devuelva los siguientes datos:

- La longitud de una cadena (sin utilizar la función strlen).
- La cantidad de vocales.
- La cantidad de palabras.

Problema 3: Tiro al blanco

a) En una Kermés se presenta el juego de tiro al blanco con dardos. El costo de un tiro es de \$100. Los premios según la puntería son:

- El círculo rojo con valor 10 tiene un premio de \$500.
- El círculo de valor 9 tiene un premio de \$200.
- El círculo de valor 8 recupera los \$100.
- El círculo de valor 7 recupera \$50.
- El resto de círculos no tiene premio.



PROGRAMACIÓN

Programador Universitario - Licenciatura en Informática - Ingeniería en Informática
Facultad de Ciencias Exactas y Tecnología - UNT

Realice un programa que permita ingresar cuantos tiros va a realizar el jugador. Luego, simule los tiros. Por último, muestre cuánta plata gastó, cuánta plata ganó y el saldo final del jugador.

Utilice las siguientes funciones:

void cargarTiros(int *puntos, int tiros); //Función para cargar los puntajes de los tiros.
int calcularGanancia(int *puntos, int tiros); //Función que calcula cuánto ganó el jugador.

b) Modifique el programa anterior para que funcione con un número indeterminado de usuarios. Además de mostrar los resultados de cada jugador, al finalizar el juego para todos los usuarios, muestre el monto total pagado por los mismos, el monto total en premios y el saldo final. Elija la marca final que considere apropiada.

Para esto modifique la función **calcularGanancia** con el siguiente prototipo:

void calcularGanancia(int *puntos, int tiros, int *totalPagos, int *totalPremios)

donde:

- **totalPagos:** acumula el pago efectuado por los jugadores
- **totalPremios:** acumula la cantidad de premios ganados por los jugadores

Problema 4: Las notas del profe

a) Realice un programa que permita calcular el promedio de notas de los alumnos de un curso.

La cantidad de alumnos dependerá del profesor que cargue las notas (según la cantidad de alumnos que posea en su materia).

Al iniciar el programa debe preguntar cuántos alumnos tiene. Simule las notas con la función `rand()` con números aleatorios de 0 a 10 y con 2 cifras decimales.

Calcule y muestre el promedio del curso y la mejor nota.

Utilice los prototipos de las siguientes funciones:

void cargarNotas(float *notas, int alumnos) // Función para simular las notas
float calcularPromedio(float *notas, int alumnos, float *promedio); // Función que calcula y guarda el promedio, además retorna la mejor nota.

b) Modifique el programa de manera tal que permita cargar las notas de los alumnos de las materias: (0)Cálculo, (1)Álgebra, (2)Programación. Cada materia puede tener diferente cantidad de alumnos. Muestre el promedio de notas y la nota más alta de cada materia. Por último muestre la materia con mejor promedio y la materia que tiene la mayor nota.

Nota: cree arreglos dinámicos para las notas de cada materia.

Guarde en un arreglo las notas promedio y en otro arreglo la mayor nota de cada materia.

Trate de analizar y utilizar las siguientes funciones:

void cargarNotas(float *notas, int alumnos);
void calcularPromedio(float *notas, int alumnos, float *promedio, float *mayorNota);
void mejores(float *promedio, float *mayorNota, int materias);