Taller de Lenguajes I – 2020
Programador Universitario / Licenciatura en informática
Trabajo Práctico N° 2

OBJETIVOS

- Utilizar punteros para manejar arreglos bidimensionales y estructuras.
- Aplicar aritmética de punteros.
- Manejar arreglos dinámicos.
- Manejar diferentes branch en git
- Llevar a cabo la codificación de programas utilizando estructuras como tipo de datos.

Ejercicios:

1) Si todavía no creó el repositorio, copie el siguiente enlace en su navegador: https://classroom.github.com/a/4P-EJHiz esto creará el repositorio para poder subir el **Trabajo Práctico Nro. 2**, realice los pasos ya aprendidos para *clonar* el repositorio en su máquina y poder comenzar a trabajar de forma *local*.

   Nota. No se olvide de incluir el archivo .gitignore en la raíz del repositorio para excluir los archivos .exe, .obj y .tds del mismo

2) En el siguiente código se accede a los elementos de una matriz.

```
#define N 4
#define M 5
Int f,c;
Double mt[N][M];
…
for(f = 0;f<N; f++)
{
      for(f = 0;f<N; f++)
       {
       cprintf("%lf    ", mt[f][c]);
       }
  cprintf("\n");
}
```

   a) Escriba el código anterior en un archivo nuevo que se llame tp2_1_1.cpp y agregue el archivo al su repositorio local y luego al repositorio remoto.

   b) Cree un nuevo Branch de forma local llamado Opcion_2 para ello utilice el comando

      git branch *Opcion_2* ← crea un nuevo branch

      git checkout *Opcion_2* ← Pone el branch [Opcion_2] como directorio de trabajo

   c) Dentro del Branch Opcion_2 cree un nuevo archivo que se llame tp2_1_2.cpp. Para asegurarse que está trabajando en el branch correspondiente ejecute el comando **git status** y lea la información que le da.

   d) En el archivo tp2_1_2.cpp escriba el código anterior, pero con **aritmética de punteros para recorrer la matriz.**

      Suba los cambios al repositorio local y remoto.

   e) Salte a la linea *master* utilizando el comando **git checkout** *master*

   f) Vaya a la carpeta donde inicializó el repositorio:

      - ¿Puede ver el archivo tp2_1_2.cpp? ¿Por qué?

g) En la línea principal agregue un nuevo archivo que se llame Respuestas.txt y responda a las preguntas anteriores.

h) Utilice el comando **gitk –all** y observe los cambios

i) Cambie al Branch *master* y vuelva a ejecutar el comando gitk –all ¿Qué diferencias nota?

j) En el Branch *master* se va a combinar (*merge)* ambos repositorios. Para esto, vamos a realizar los siguientes pasos:

- **Git checkout** *master* ← nos movemos al branch *master*
- **git merge** *Opcion_2* ← combinamos *master* con *Opcion_2* y lo dejamos en *master*
- Luego utilice el comando **gitk –all** para ver los cambios, ¿Qué nota?
- Realice el push para llevar sus cambios al repositorio Remoto

Sobre la línea principal:

3) Escribe un programa en el que se cargue una matriz de 15 filas con números enteros aleatorios entre 100 y 999. Cada fila se cargará en forma dinámica. La cantidad de columnas estará dada por un número aleatorio entre [5, 15]. Mostrar por pantalla la matriz.
   a) Luego determinar para cada fila cuántos números son pares.
   b) Por último, generar un nuevo **vector dinámico** de 15 elementos con la cantidad de pares por fila obtenidos en el punto anterior.
   c) Subir los cambios al repositorio remoto.

4) Declara un **tipo de dato estructura**:

d) Para representar a una PC; los campos serán: velocidad de procesamiento en GHz, año de fabricación, tipo de procesador, cantidad de núcleos.

e) Considera valores enteros aleatorios para la velocidad: entre 1 y 3, para el año: entre 2000 y 2017, para la cantidad de núcleos: entre 1 y 4.

f) Para evitar ingresar por teclado los tipos de procesador, considera que estos se encuentran en un arreglo de cadenas de caracteres:

```
char tipos[6][10]={“Intel”, “AMD”, “Celeron”, “Athlon”, “Core”, “Pentium”}
```

La **estructura** será la siguiente:

```
struct compu {
        int velocidad;//entre 1 y 3 Gherz
        int anio;//entre 2000 y 2017
        int cantidad;//entre 1 y 4
        char *tipo_cpu;//valores del arreglo tipos
};
```

g) Define una variable del tipo arreglo de estructura para guardar las características (punto a) de la cantidad de PC que ingrese el usuario (arreglo dinámico).

h) Escribe una función que devuelva una PC pasando la estructura como argumento a la **función**.

i) Escribe una función a la que se le pase el **puntero al arreglo de estructuras** y cargue el mismo con los valores asignados a las características de cada PC.

j) Escribe una función que presente la lista de las PC, cada una con sus características.

k) Escribe una función que presente la PC más vieja.

l) Escribe una función que presente la PC que tiene mayor velocidad.