

**APELLIDO Y NOMBRE:** \_\_\_\_\_ **CANTIDAD DE HOJAS:** \_\_\_\_\_

**PARTE I – Complete el programa ( 26 puntos)**

1.1 Completar el código de manera tal que la función genere un archivo con los datos ingresados por teclado.

```
void GenerarArchivo(char *archivo)
{
    FILE *ptrArchivo;
    real numero;
    int entero, decimal;
    _____;

    if ( _____ )
    { do
        { printf("Ingrese valor entero, Finaliza con Valor 0 (Cero)\n");
          scanf("%d", &entero);
          if ( _____ )
          { printf("Ingrese valor decimal\n");
            scanf("%d", &decimal);
            numero.entero=entero;
            numero.decimal=decimal;
            _____;
          }
        } while ( _____ );
    }
    _____;
}
```

```
typedef struct
{
    int entero;
    int decimal;
}real;
```

**PARTE II – Opción múltiple (26 puntos, solamente si están todas correctas)**

2 - Marque la opción correcta que se corresponda con lo que el código imprimiría por pantalla:

```
char x (char a)
{
    if (a<'b')
        printf("%c", x(a+1));
    return 'a';
    if (a>='b')
        return 'b';
}
void main()
{
    printf("%c",x('a'));
}
(A) aaa (B) ab (C) aabb (D) aa (E) bbb
(F) aabaa (G) bbaa
```

```
int main()
{
    char nombre[]="Fernando";
    recursiva(nombre, 1);
    printf("%s",nombre);
}
void recursiva(char *nombre, int i)
{
    char aux;
    if(*nombre!='\0' )
    {
        recursiva(nombre+1, i+1);
        if(*nombre>'a' && *nombre<'z')
        { *nombre=65; }
        else
        { *nombre=*nombre&255; }
    }
}
(A) fernando (B) FAAAaAAA (C) 6565656565
(D) fERNANDO (E) Fernando (G) fErNaNdO
```

```
void main()
{
    char b[] = "3456789";
    char * p = b+5 ;
    int i=0;
    while(*(b+i)!='\0')
        i++;
    while(i>0 && *p<'9')
    {
        i--;
        *p=*p+1;
    }
    printf( "%s%s", p,b );
}
(A) 34567893456789 (B) 343456799 (C)
54567893456789 (D) 993456799 (E) 3456789
(F) 3456789: 3456789
```

```
unsigned char a;
a = ( ~ ((31<<1)) & 127);
printf("%d",a);
(A) -65 (B) 1 (C) 9 (D) 15 (E) 3 (F) 65 (G) 34
```

**APELLIDO Y NOMBRE:** \_\_\_\_\_ **CANTIDAD DE HOJAS:** \_\_\_\_\_

---

PARTE III – Programación (48 puntos, 24 cada ejercicio)

Archivo	Contenido
Alumnos.txt	<u>Registros:</u> NroRegistro (int) Nombre (int) PromedioGral (Double)
Materias.dat	<u>Registros:</u> NroRegistro (int) NombreMateria (char [250]) Notas (unsigned int [10])

- 1- Realizar una función que reciba como parámetro dos nombres de archivos (Alumnos.txt y Materias.dat) y que calcule el promedio general del alumno y lo vuelva a almacenar en el archivo recibido como primer parámetro.
- 2- Realizar una función que reciba como parámetro un string (NOMBRE DE MATERIA) y retorne un árbol binario de búsqueda (ABB) cargado con todos los alumnos que rindieron por lo menos dos exámenes en dicha materia, el árbol debe estar ordenado por Nro. de Registro.

---

Condiciones para todos los ejercicios:

- Para aprobar el examen se debe sumar un mínimo de 65 puntos.
  - Declarar todos los "struct" y "typedef" que utilice para resolver el ejercicio.
  - No desperdiciar memoria. Liberar todos los espacios que se hayan reservado y que no se utilicen.
  - Se pueden programar funciones adicionales, respetando los puntos anteriores.
  - Considere que las estructuras recibidas como parámetro por las funciones pueden estar vacías (nulas).
  - Las pilas y colas deben respetar el protocolo y el método elegido para su implementación a lo largo de todo el examen.
-