1. Una empresa debe realizar el pago de bono de fin de año para los empleados que hayan logrado al menos 5 objetivos propuestos para el año en curso.
   El programa que administra al personal posee un archivo "empleados.data" con los datos de todos los trabajadores. Cada bloque es un:

```
struct Empleado
{
int nLegajo; //número de lejago para identificar a cada empleado
char[50] nombres;
char[50] apellidos;
long dni;
int edad;
int objetivos_logrados; //este año
        bool recibeBono = false;
};
```

Además, el software posee una lista DE de ex empleados con el nLegajo y fecha de baja (AAAAMMDD) ordenados ascendentemente por nLegajo.

a) Declare el tipo NODO de la lista de ex empleados y todos los tipos que se utilicen dentro del nodo. (1PTS)

b) Implemente un procedimiento con el siguiente encabezado:

void pagarBono(char arch_empleados[], Nodo *listaBajas)

y que establezca en true el miembro recibeBono a todos los empleados que hayan cumplido con la regla de los objetivos y no estén en la lista de bajas. Puede utilizar el subprograma **ya desarrollado:**

Nodo *bucarBaja(Nodo *listaBajas, int nLegajo)

Que recibe la lista de bajas y un número de legajo, busca al empleado en la lista de bajas y, si lo encuentra devuelve un puntero al nodo correspondiente, caso contrario devuelve NULL.
(Aclaración: Todo cambio debe ser guardado en el archivo "empleados.data", puede desarrollarse. 3 PTS)

c) Desarrolle un subprograma que cumpla:

void agregarBaja(?, int nLegajo, int fechaBaja)

que inserte ordenadamente por número de legajo a un empleado que ha sido dado de baja.
En la ubicación del símbolo "?" debe pasarse como parámetro la lista de bajas. Complete correctamente e Indique el método de pasaje de parámetros adecuado para este caso, justifique la respuesta (3 PTS).

2.

```
void func_ej_1(int *x, int pos, int valor)
{
     *(x+pos) = valor;
     return;
}

int main()
{
     int *p = NULL;
     int x[] = {21, 22, 23, 24, 25};
     p = x;
     func_ej_1(p,2,500);
     func_ej_1(p,4,500);
     for (int i=0;i<5; i++)
          cout << p[i] << "  ";
     return 0;
}
```

a) Indicar la salida por pantalla del programa

b) ¿Es correcto que no haya una instrucción delete antes de finalizar? Justificar la respuesta.

3.

```
struct Nodo
{
          int info;
          Nodo *sgte;
};

int main()
{
     Nodo *p= NULL;
     Nodo *aux;
     p = new Nodo();
     p->info = 15;
     p->sgte= new Nodo();
     p->sgte->info = 20;
     p->sgte->sgte = NULL;
     aux = p->sgte;
     p->sgte = new  Nodo();
     p->sgte->info = 90;
     p->sgte->sgte= aux;
     aux=p;
     while (aux)
     {
          cout << aux->info << "; ";
          aux= aux->sgte;
     }
     return 0;
}
```

Este programa muestra (seleccione la opción correcta):
90; 15; 20;
15; 90; 20;
15; 20; 90;
15; 90;
20; 90;
Nada
No compila JUSTIFICAR.

4.

```
FILE *f;
int y,z;
f = fopen("unarchivo","wb");
int vec[] = {1,2,3,4,5,6,7,8,9,10};
fwrite(vec,sizeof(int),10,f);
fclose(f);
f = fopen("unarchivo","rb+");
fread(&y,sizeof(int),1,f);
fread(&y,sizeof(int),1,f);
fread(&y,sizeof(int),1,f);
y += 1000;
fseek(f, (-1)*sizeof(int),SEEK_CUR);
fwrite(&y,sizeof(int),1,f);
fseek(f, 0,SEEK_SET);
while(fread(&z,sizeof(int),1,f))
{
    cout << z << ";";
}
fclose(f);
```

Indicar la salida por pantalla del programa: