UTN FRBA - Algoritmos y Estructura de Datos – Comisión Ing. Pablo Mendez

Curso: Apellido y Nombres: Legajo: Fecha:

Aclaración: Puede asumir en todos los casos que el tipo “Nodo” es el declarado en el punto 2.

1)

```
int *x; int *y;
int w;
x = new int();
w = 15;
*x = w;
y = x;
w++;
cout << *y ;
```

La salida por pantalla es:

a) 16
b) 14
c) 15
d) Una dirección de memoria
e) Ninguna, no compila. JUSTIFICAR.

2)

```
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:

a) 90; 15; 20;
b) 15; 90; 20;
c) 15; 20; 90;
d) 15; 90;
e) 20; 90;
f) Nada
g) No compila JUSTIFICAR.

3)

```
void insertar(Nodo *&l, int x)
{
    Nodo *paux = l;
    Nodo *paux2;
    if (l)
    {
        if (paux->info < x)
        {
                while(paux->sgte && paux->sgte->info < x)
                    paux = paux->sgte;
                paux2= new Nodo();
                paux2->sgte= paux->sgte;
                paux2->info= x;
                paux->sgte = paux2;
        }
        else
        {
                paux2= new Nodo();
                paux2->sgte= paux;
                paux2->info= x;
                l = paux2;
        }
    }
    else
    {
        l= new Nodo();
        l->info=x;
        l->sgte=NULL;
    }
    return;
```

```
int main()
{
    Nodo *p= NULL; Nodo *aux;
    insertar(p,76); insertar(p,34);
    insertar(p,92); insertar(p,5);
    aux=p;
    while (aux)
    {
        cout << aux->info << "; ";
        aux= aux->sgte;
    }
    return 0;
}
```

Este programa muestra:

a) 76; 34; 92; 5;
b) 34; 5; 76; 92;
c) 5; 34; 76; 92;
d) 76;
e) No muestra nada. Explicar por qué.
f) No compila. JUSTIFICAR.

UTN FRBA - Algoritmos y Estructura de Datos – Comisión Ing. Pablo Mendez

Curso: Apellido y Nombres: Legajo: Fecha:

4)

```
int *p; int tam;
cout << "Ingrese un tamaño" << endl;
cin >> tam;
p = new int[tam + 5];
p[0] = 67;
p[1] = 72;
p[2] = 27;
p[3] = 81;
p[4] = 23;
cout << *(p+2) << endl;
return 0;
```

Este programa muestra:

a) 67
b) 72
c) 27
d) Una dirección de memoria.
e) Compila pero tiene comportamiento indeterminado. JUSTIFICAR.
f) Nada, no compila. JUSTIFICAR.

5)

```
int main()
{
    FILE *f;
    int x;
    int y = 555;
    f = fopen("prueba.laquequieras","wb");
    int vec[] = {91,92,93,94,95,96,97,98,99,100};
    fwrite(vec,sizeof(int),10,f);
    fclose(f);
    f = fopen("prueba.laquequieras","rb+");
    fseek(f, 0,SEEK_SET);
    fread(&x,sizeof(int),1,f);
    fread(&x,sizeof(int),1,f);
    fread(&x,sizeof(int),1,f);
    fseek(f, (-1)*sizeof(int),SEEK_CUR);
    fwrite(&y,sizeof(int),1,f);
    fseek(f, 0,SEEK_SET);
    fread(&x,sizeof(int),1,f);
    while(!feof(f))
    {
        cout << x << ";";
        fread(&x,sizeof(int),1,f);
    }
    fclose(f);
}
```

Este programa muestra:

a) 91;92;93;
b) 91;92;93;94;95;96;97;98;99;555;
c) 91;555;93;94;95;96;97;98;99;100;
d) 91;92;93;555;95;96;97;98;99;100;
e) 91;92;555;94;95;96;97;98;99;100;
f) 91;92;93;555;94;95;96;97;98;99;100;
g) No compila, JUSTIFICAR.

6)

```
void doThat(int *&x)
{
    x= new int();
    *x= 5;
}

int main()
{
    int x;
    x = 40;
    int *p;
    p = &x;
    x++;
    doThat(p);
    cout << *p;
    return 0;
}
```

Este programa muestra:

a) 40
b) 41
c) 5
d) 6
e) Una dirección de memoria
f) Nada, no compila JUSTIFICAR.

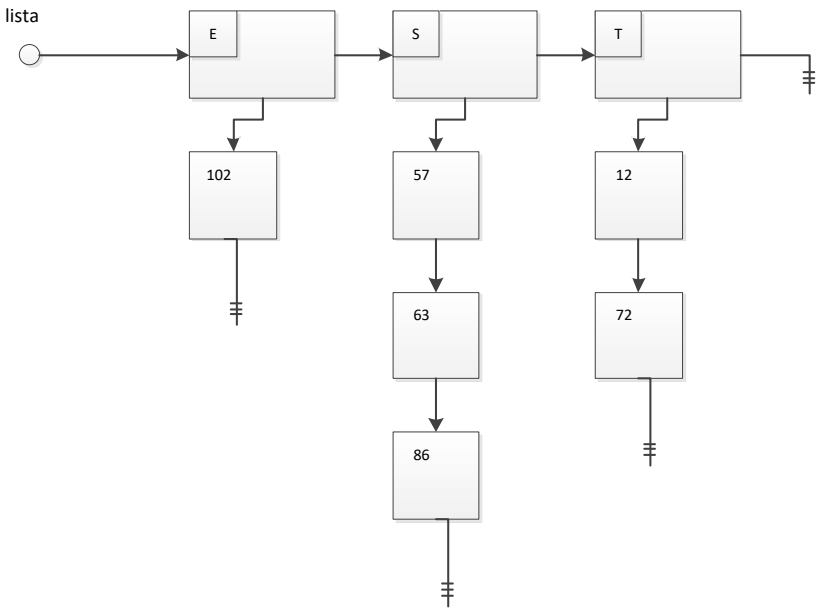
7) Teniendo en cuenta que:

```
struct NodoNumero
{
    int info;
    NodoNumero *sgte = NULL;
}

struct NodoLetra
{
    char info;
    NodoLetra *sgte = NULL;
    NodoNumero *sublista = NULL;
}


NodoLetra *lista = NULL;
```

Y luego de un tiempo la lista se encuentra en este estado:



Realice un código que inserte el elemento “72” ordenadamente en la sublista del elemento “S”. (sólo una porción de código, no hace falta que sea un subprograma ni un main, se está evaluando el tema “punteros”).

Curso: Apellido y Nombres: Legajo: Fecha:

8) Se tiene un tipo de datos *Empleado* (detallado al final del enunciado), y dos archivos binarios, llamados "cadetes.bat" y "jefes.bat", en los cuales se guardan, *uno tras otro*, los datos de los empleados de una empresa. Desarrolle un procedimiento que cumpla el siguiente prototipo:

void promocionar(int n, char arch_jefes[], char arch_cadetes[])

que permita "promocionar" a un cierto empleado (cuyo nOrden sea el parámetro pasado, es decir, cuyo ordinal sea n), pasándolo de cadete a jefe.
NOTA: El pasaje implica moverlo de un archivo a otro; no es necesario borrar el registro de un cadete al promocionarlo y convertirlo en jefe, solo importa que quede registrado en el archivo de jefes.
A su vez, **no deben borrarse los jefes ya registrados al querer insertar uno nuevo**.

```
struct Empleado
{
        int nOrden;             //Ordinal para identificar a cada empleado
        char[20] nombres;
        char[30] apellidos;
        long dni;
        int edad;
};
```