

Por favor siga las siguientes indicaciones:

- Escriba con lápiz.
- Escriba su nombre y número de documento en todas las hojas que entregue.
- Numere las hojas e indique el total de hojas en la primera de ellas.
- Escriba las hojas de un solo lado.
- Comience cada ejercicio en una hoja nueva.
- El total máximo de puntos del examen es 100.
- El examen contiene un total de: 2 páginas.
- La prueba es individual y sin material.
- Solo se contestan dudas acerca de la letra de los ejercicios.
- Duración 3 horas.

Ejercicio 1 (15 puntos)

1. *¿Qué sucede si pasamos un parámetro a una función por referencia y lo modificamos dentro de ella? ¿Cómo lo hacemos?*

Solución:

Si pasamos un parámetro a una función por referencia los cambios que le realicemos dentro de la función se van a ver reflejados fuera de la misma. Lo hacemos agregando & entre el tipo del parámetro y el nombre. Por ejemplo:

```
void ejemplo(int inc, int & valor){
    valor = valor + inc;
}

int main(){

    int valor = 7;
    int inc = 3;
    printf("inc=%d, valor=%d\n",inc, valor);
    ejemplo(inc, valor);
    printf("inc=%d, valor=%d\n",inc, valor);

    return 0;
}
```

Salida:

inc=3, valor=7
inc=3, valor=10

La función ejemplo tiene dos parámetros, uno por valor "inc" y otro por referencia "valor". Los cambios que se realicen al parámetro valor dentro de la función ejemplo se ven reflejados en la variable utilizada para la invocación en el main. No ocurre así con el parámetro inc, ya que como se pasa por valor, la función ejemplo trabaja con una copia del mismo.

2. *Describe (no implemente) el algoritmo de ordenación por selección (Selection Sort).*

Solución:

La idea del selection sort es la siguiente: en la pasada i -ésima seleccionamos el elemento menor entre $A[i]; \dots; A[n]$ y lo intercambiamos con el $A[i]$. Como resultado, luego de i pasadas los menores i elementos ocuparán las posiciones $A[1]; \dots; A[i]$ y además los elementos de dichas posiciones estarán ordenados.

Ejercicio 2 (30 puntos)

Una palabra es una secuencia ordenada de caracteres que puede ser vacía y que nunca puede tener más de 50 caracteres. Por tanto la estructura que modela la palabra podría ser la siguiente:

```
#define largo 50

struct palabra{
    char elems[largo];
    int pos;
};
```

Se utilizará el campo `pos` para indicar el último carácter en la palabra. Si la palabra es vacía `pos = -1`.

Ejemplo: Si palabra es “Hola” entonces `elems = ['H','o','l','a']` y `pos=3`

Se pide implementar las siguientes funciones.

1. `bool esPrefijo (palabra p1, palabra p2);`
Determina si la palabra `p1` es prefijo de la palabra `p2`.
`p1` será prefijo de `p2` si y solo si `p1` es un segmento inicial de `p2`.
La palabra vacía es prefijo de todas las palabras.

Ejemplo: “Buen” es prefijo de “Bueno” pero no de “Bulto”.

Solución:

```
bool esPrefijo(palabra p1, palabra p2){

    bool esPrefijo = false;

    if (p1.pos <= p2.pos){
        int i = 0;
        while ((p1.elems[i]== p2.elems[i])&& (i<=p1.pos)){
            ++i;
        }
        if (i > p1.pos)
            esPrefijo = true;
        else esPrefijo = false;
    }
    return esPrefijo;
}
```

2. `int lengthSinCar (palabra pal, char car);`

Calcula la longitud de una cadena de texto, sin considerar al carácter `car`.

Si la cadena es vacía devuelve 0.

Ejemplo ilustrativo: `lengthSinCar ("Cocina", "c")= 4`

`lengthSinCar ("Cocina", "a")= 5`

Solución:

```
int lengthSinCar (palabra pal, char car){  
  
    int cant = 0;  
    int i = 0;  
  
    while (i <= pal.pos){  
        if (pal.elems[i] != car){  
            cant ++;  
        }  
        i++;  
    }  
}
```

Ejercicio 3 (15 puntos)

Implemente recursivamente la función *ultimaAparicion*, a la cual se le pasa por parámetro una palabra, el largo de la misma, y un carácter, y devuelve la posición en la palabra donde aparece por última vez el carácter pasado por parámetro. Si el carácter no aparece en la palabra, la función debe devolver -1.

Usar el siguiente cabezal para la función: *int ultimaAparicion(char str[], int largostr, char car)*

Ejemplos: *ultimaAparicion("Mama", 4, 'a')= 3*
ultimaAparicion("Mama", 4, 'm')= 2
ultimaAparicion("", 0, 'i')= -1
ultimaAparicion("estudiar", 8, 'z')= -1

Solución:

```
int ultimaAparicion(char str[], int largostr, char car){
    if (largostr==0){
        return -1;
    }
    else if (str[largostr-1]==car)
        return largostr-1;
    else return ultimaAparicion(str, largostr-1, car);
}
```

Ejercicio 4 (40 puntos)

La cadena de supermercados "COMPRES BARATOS", va a sortear 15 automóviles entre sus clientes (1 por día) y una vez finalizado cada sorteo, desea registrar los resultados del mismo. Para cada sorteo necesita registrar la fecha en que se realizó, y los siguientes datos del ganador: cédula, apellido y nombre.

Escribe un programa en C/C++ que despliegue y maneje el siguiente menú:

- 1- Alta de resultado de sorteo
- 2- Listar resultados de sorteos
- 3- Buscar resultado de sorteo
- 0- Salir

En la **opción 1** se le pedirá al usuario la fecha del sorteo y los datos del ganador, y se le dará de alta en el sistema, siempre y cuando no exista ya un sorteo registrado para esa fecha.

En la **opción 2** se deben desplegar en la pantalla todos los datos de los sorteos realizados (fecha y datos del ganador).

En la **opción 3** dada la fecha de un sorteo, se buscará en los sorteos realizados, y si ya existe un sorteo llevado a cabo en esa fecha, deberá mostrar los datos del ganador, en caso contrario se deberá mostrar "Sorteo no registrado".

En la **opción 0** el programa debe terminar.

Notas:

- La cédula se representará con un arreglo de caracteres de largo 8.
- El apellido y el nombre se representarán como arreglos de caracteres de largo 20.

- La fecha del sorteo se representará con una estructura con 3 campos de tipo entero (un campo para el día, otro para el mes, y otro para el año).
- Se pueden definir funciones y procedimientos auxiliares.

Solución:

```
#include <cstdlib>
#include <iostream>
using namespace std;

#define MAX_LARGO_CEDULA 8
#define MAX_LARGO_NOMBRE 20
#define MAX_CANT_SORTEOS 15

//estructura para guardar una fecha
struct Fecha{
    int dia;
    int mes;
    int ano;
};

//estructura para guardar los datos de un sorteo
struct S_sorteo{
    Fecha fechaRealizado;
    char cedula[MAX_LARGO_CEDULA];
    char nombre[MAX_LARGO_NOMBRE];
    char apellido[MAX_LARGO_NOMBRE];
};

//array con tope para guardar los resultados de los sorteos
struct Sorteos{
    S_sorteo resultados[MAX_CANT_SORTEOS];
    int tope; //cantidad de sorteos realizados
};

bool igualesFechas(Fecha f1, Fecha f2){

    bool iguales;

    if((f1.dia == f2.dia)&&(f1.mes == f2.mes)&&(f1.ano == f2.ano))
        iguales = true;
    else iguales = false;
    return iguales;
}

//Busca si existe un sorteo en la fecha pasada por parametro
//Si existe devuelve la posicion del mismo en el arreglo, si no existe devuelve
-1
int buscarResultadoSorteo(Sorteos sorteosRealizados, Fecha fechaSorteo){

    int pos = 0;
    bool encuentre = false;

    while (!encontrer && (pos <= sorteosRealizados.tope-1)){
        if (igualesFechas(sorteosRealizados.resultados[pos].fechaRealizado,
fechaSorteo))
            encuentre = true;
        else ++pos;
    }
}
```

```
    if (encontre)
        return pos;
    else return -1;
}

int altaResultadoSorteo(Sorteos & sorteosRealizados, S_sorteo sorteo){

    int res = -1;

    if (buscarResultadoSorteo(sorteosRealizados,sorteo.fechaRealizado)==-1){
        //no existe sorteo para esa fecha, por lo que lo paso a agregar
        int posLibre = sorteosRealizados.tope;
        sorteosRealizados.resultados[posLibre].fechaRealizado.dia =
sorteo.fechaRealizado.dia;
        sorteosRealizados.resultados[posLibre].fechaRealizado.mes =
sorteo.fechaRealizado.mes;
        sorteosRealizados.resultados[posLibre].fechaRealizado.ano =
sorteo.fechaRealizado.ano;

        strcpy(sorteosRealizados.resultados[posLibre].cedula, sorteo.cedula);
        strcpy(sorteosRealizados.resultados[posLibre].nombre, sorteo.nombre);
        strcpy(sorteosRealizados.resultados[posLibre].apellido, sorteo.apellido);

        ++ sorteosRealizados.tope;//aumento la cantidad de sorteos
        res = posLibre;
    }

    return res;
}

void listarResultadosSorteos(Sorteos sorteosRealizados){

    if (sorteosRealizados.tope == -1)
        printf("No existen sorteos.\n");
    else {
        int pos = 0;
        while(pos <= sorteosRealizados.tope -1){
            printf("---Datos de sorteo %d\n", pos+1);
            printf("Fecha realizado:
%d/%d/%d\n",sorteosRealizados.resultados[pos].fechaRealizado.dia,sorteosRealizad
os.resultados[pos].fechaRealizado.mes,sorteosRealizados.resultados[pos].fechaRea
lizado.ano);
            printf("Cedula del ganador:
%s\n",sorteosRealizados.resultados[pos].cedula);
            printf("Nombre del ganador:
%s\n",sorteosRealizados.resultados[pos].nombre);
            printf("Apellido del ganador:
%s\n",sorteosRealizados.resultados[pos].apellido);
            ++pos;
        }
    }
}

int main()
{

    Sorteos sorteosRealizados;
    sorteosRealizados.tope = 0;

    int opcion;
    Fecha fechaSorteo;
```

```
S_sorteo sorteo;
int res;

do {
    printf("Menu de opciones\n");
    printf("1- Alta de resultado de sorteo\n");
    printf("2- Listar resultados de sorteos\n");
    printf("3- Buscar resultado de sorteo\n");
    printf("0- Salir\n");
    printf("Ingrese la opcion deseada: ");
    scanf("%d",&opcion);

    switch (opcion){
        case 1:
            printf("Ingrese la fecha del sorteo en el formato
dia/mes/ano: ");

scanf("%d/%d/%d",&sorteo.fechaRealizado.dia,&sorteo.fechaRealizado.mes,&sorteo.f
echaRealizado.ano);

            printf("Ingrese la cedula del ganador (sin digito
verificador): ");
            scanf("%s",sorteo.cedula);
            printf("Ingrese el nombre del ganador: ");
            scanf("%s",sorteo.nombre);
            printf("Ingrese el apellido del ganador: ");
            scanf("%s",sorteo.apellido);

            res = altaResultadoSorteo(sorteosRealizados,sorteo);

            if (res== -1)
                printf("Ya existe un sorteo registrado en esa fecha.\n");
            else printf("Sorteo numero %d agregado con exito.\n",res+1);

            break;
        case 2:
            listarResultadosSorteos(sorteosRealizados);
            break;
        case 3:
            printf("Ingrese la fecha del sorteo a buscar en el formato
dia/mes/ano: ");

scanf("%d/%d/%d",&fechaSorteo.dia,&fechaSorteo.mes,&fechaSorteo.ano);
            res = buscarResultadoSorteo(sorteosRealizados,fechaSorteo);

            if (res == -1){
                printf("Sorteo no registrado.\n");
            }
            else {
                printf("Datos del ganador:\n");
                printf("Cedula:
%s\n",sorteosRealizados.resultados[res].cedula);
                printf("Nombre:
%s\n",sorteosRealizados.resultados[res].nombre);
                printf("Apellido:
%s\n",sorteosRealizados.resultados[res].apellido);
            }
            break;
    }
}while (opcion!=0);

system("PAUSE");
return EXIT_SUCCESS;
}
```