

Universidad Francisco Marroquín
F.I.S.I.C.C.
Programación III
1er semestre 1992
Prof. : Egdare Futch H.

Segundo Examen Parcial

Instrucciones

Este segundo examen parcial cubre lo visto en clase y lo implementado durante los proyectos. Debe contestar con respuestas directas, claras y legibles. Sus programas deben estar correctos, tanto a nivel de sintaxis como de uso de procedimientos y funciones. Buena suerte !!!

cryin' won't help you,
prayin' won't do ya no good.
- Led Zeppelin

Pregunta #1 (30 puntos)

a) Construya un container class abstracto llamado `Array`, bajo el cual se puedan crear arrays de tipo Pascal : con límite inferior distinto de cero, límite superior diferente al tamaño del array y chequeo de rango (imprimir en cerr "Range error with subscript n "). Nótese que debe sobrecargar el operador `[]`.

b) Por medio de herencia, construya una clase concreta, derivada de `Array`, que contenga enteros. Uno de los requerimientos más importantes es que **no debe desperdiciar espacio**. Debe ser posible hacer algo como :

```
#include "intarray.h"
main()
{
    IntArray a(10,20);        // array[10..20]

    for (h = 10 ; h < 22 ; h++)
        a[h] = h;           // error en cuanto h == 21
}
```

Pregunta #2 (10 puntos)

Modifique el siguiente programa :

```
#include <iostream.h>

main()
{
    cout << "Hello, world!" << endl;
}
```

para que imprima

```
Start
Hello, world!
End
```

sin modificar `main()`.

Pregunta #3 (20 puntos)

Construya en C++ una función que tome como parámetros un `int` indicando cuántos strings vienen después y n strings. Esta función debe devolver un pointer a la concatenación de todos estos strings. Ejemplo :

```
newstr = concat(3,"hola"," como"," estan"); // newstr = "hola como estan"
new2str = concat(2,str1,str2);                // new2str = str1+str2
```

Pregunta #4 (20 puntos)

Se desea hacer un programa que tenga dos procedimientos, uno llamado `producer()` y otro llamado `consumer()`, en el cual la primera rutina haga un push de un `int` aleatorio a una instancia de `IntStack`. Por medio de un salto no local, se transfiere control a la segunda rutina, la cual deberá hacer un pop del stack, imprimir el número y regresar, por medio de otro salto no local a la primera rutina. Esta interacción entre rutinas debe mantenerse hasta que la rutina `consumer()` encuentre el número 999. Implemente este programa, asumiendo que el `IntStack` está ya implementado en algún otro módulo.

Pregunta #5 (20 puntos)

Implemente el operador `BigNum::++(void)` con todas sus rutinas auxiliares.