Examen Final de Ciencia de la Computación II

Grupo: CCOMP3-1/ CCOMP3-2

1. (Habilidad A,B )Dado el siguiente código en C++ (5 puntos)

char \* str = "Your dog has fleas.";

char \* buf0 ;

int main(){

char \* str2 = "Don't make fun of my dog." ;

static char \* str3 = str;

char \* buf1 ;

buf0 = malloc(BUFSIZ);

buf1 = malloc(BUFSIZ);

return 0;

}

Indique en que segmento de la memoria de la aplicación se localizan, el puntero y el valor que apunta.

|  |  |  |
| --- | --- | --- |
| **Puntero** | **Ubicación de la Variable Puntero** | **Ubicación del Valor apuntado** |
| str |  |  |
| str2 |  |  |
| str3 |  |  |
| buf0 |  |  |
| buf1 |  |  |

1. (Habilidad A,B) Escriba los valores impresos por el siguiente programa (5 puntos)

class operacion

{

private:

int \_x;

public:

operacion(int x):\_x(x) {};

int operator()(int a)

{ return \_x+a;}

};

template<typename T>

void operador(T a, int \*b)

{

\*(b-1) = \*b;

\*b = a(\*b);

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

int \*a = new int[2];

\*a = 0;

\*(a+1) = 1;

for(int i= 0; i<=10; i++)

{

cout<<\*a<<endl;

operador(operacion(\*a),a+1);

}

system("PAUSE");

return 0;

}

1. (Habilidad A,B) El siguiente código, es parte de la definición de una pila (5 puntos)

template <class T>

class Stack {

public:

Stack():top(0) { }

~Stack() {

while ( !isEmpty() ) {

pop();

}

isEmpty();

}

void push (const T& object);

T pop();

bool isEmpty();

private:

struct StackNode {

T data;

StackNode \*next;

StackNode(const T& newData, StackNode \*nextNode)

: data(newData), next(nextNode) {}

};

StackNode \*top;

};

template <class T>

bool Stack<T>::isEmpty() {

if (top == 0) {

return true;

}

else {

return false;

}

Realice los métodos push y pop acorde a la definición de la pila descrita líneas arriba, para ello utilice el espacio reservado para cada método en la siguiente página.

template <class T>

void Stack<T>::push(const T& obj) {

}

template <class T>

T Stack<T>::pop() {

}

1. (Habilidades A,B) Tomando como referencia el segmento de código siguiente, completar la sobrecarga del paréntesis de la clase operador y la función main, de tal forma que obtenga los mismos resultados de la pregunta 2 (5 puntos)

class operacion

{

public:

int operator()(int a, int b)

{ return a+b;}

};

template<typename T>

class operador

{

private:

T op;

public:

**void operator()(int \*a)**

**{**

**}**

};

**int \_tmain(int argc, \_TCHAR\* argv[])**

{

int \*a = new int[2];

\*a = 0;

\*(a+1) = 1;

for(int i= 0; i<=10; i++)

{

cout<<\*a<<std;

}

system("PAUSE");

return 0;

}