Pila::Pila(int max)

{ cabeza = new nodo; z=new nodo;

cabeza->siguiente=z; z->siguiente =z;

}

Pila::~Pila()

{struct nodo \*t=cabeza;

while (t!=z)

{cabeza=t; t->siguiente; delete cabeza;}

}

void Pila:: meter (tipoElemento V)

{struct nodo \*t=new nodo;

t->clave = v; t->siguiente=cabeza->siguiente;

cabeza->siguiente=t;

}

tipoElemento Pila:: Sacar()

{tipoElemento x;

struct nodo \*t = cabeza->siguiente;

cabeza->siguiente = t->siguiente; x=t->clave;

delete t; return x;

}

int Pila::vacia()

{return cabeza->siguiente == z;}

Class Pila

{public:

Pila (int max);

~Pila();

void meter(tipoElemento V);

tipoElemento sacar();

int vacìa();

prívate:

struct nodo

{tipoElemento clave; struct nodo \*siguiente;}

struct nodo \*cabeza, \*z;

};

class Pila

{ tipoElemento \*pila;

Int p;

Public:

Pila (int max=100)

{pila = new tipoElemento[max]; p=0;}

~Pila()

{delete pila;}

Inline void meter (tipoElemento v)

{pila[p++]=v;}

Inline tipoElemento sacar()

{return pila [--p];}

Inline int vacia()

{return !p;}

};

-------------

char c; Pila guardar(50);

while(cin.get(c))

{ while(c>=’0’ && c<=’9’)

{cout.put(c); cin.get(c);}

if (c==’)’) cout.put(guardar.sacar());

if (c==’+’) guardar.meter(c);

if (c==’\*’) guardar.meter(c);

if (c!=’(‘) cout << ‘ ‘;

}

cout<< ‘\n’;

char c; Pila acc(50); int x;

while(cin.get(c))

{x=0;

while (c==’ ‘) cin.get(c);

while(c>= ‘0’ && c<=’9’)

{x=10\*x + (c-‘0’); cin.get(c);}

if (c == ‘+’) x= acc.sacar() + acc.sacar();

if (c== ’\*’) x= acc.sacar() \* acc.sacar();

acc.meter(x);

}

cout << acc.sacar()<< ‘\n’;