

[Área personal](#) / [Mis cursos](#) / [\[4-2021\] INF220-SM](#) / [General](#) / [Examen Mesa Examinadora](#)**Comenzado el** Thursday, 9 de September de 2021, 07:10**Estado** Finalizado**Finalizado en** Thursday, 9 de September de 2021, 08:10**Tiempo
empleado** 59 minutos 45 segundos**Calificación** Sin calificar aún**Pregunta 1**

Finalizado

Puntúa como 1,00

Tomando en cuenta que se tiene El TDA Pila que tiene como unico atributo Tope de Tipo Entero implemente los metodos meter y sacar pertenecientes a la clase Pila. (para el usuario la pila solo puede guardar 7 elementos maximo y ademas los unicos valores permitidos son 1,2,3,4,5,6,7,8,9)

Considerando que el metodo meter tiene como parametro por valor E de tipo Entero y el metodo Sacar tiene como parametro por referencia E de tipo entero

Se hace notar que si la pila tiene los elementos 1 ,2 y 9 entonces el atributo tope tendria '129'

```
int cantidadD(int x){
    int c = 0;
    int r;
    while(x!=0){
        x = x/10;
        c++;
    }
}

void Pila::meter(int E){
    if(cantidadD(Tope)<8){
        if(Tope==0) Tope=E else Tope = Tope*10 + E;
    }
}

void Pila::sacar(int& E){
    if(!vacía()){
        E = Tope%10;
        Tope = Tope/10;
    }else{
        cout<<"ERROR: Pila vacía"
    }
}
```

Pregunta 2

Finalizado

Puntúa como 1,00

Tomando en cuenta que se tiene El TDA Cola que tiene como unico atributo S de Tipo Entero implemente los metodos poner y sacar pertenecientes a la clase Cola.(para el usuario la cola solo puede guardar 7 elementos maximo y ademas los unicos valores permitidos son 1,2,3,4,5,6,7,8,9)

Considerando que el metodo poner tiene como parametro por valor E de tipo Entero y el metodo Sacar tiene como parametro por referencia E de tipo entero

Se hace notar que si la cola tiene los elementos 1 ,2 y 9 entonces el atributo S tendria **'129'**

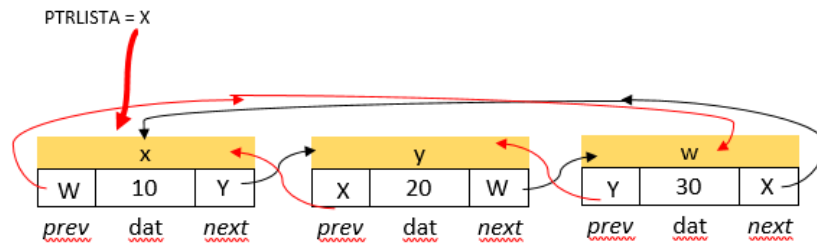
```
void Cola::poner(int E) {  
    if(cantidadD(S)<7 && E>0 && E<19){  
        if(vacia()){  
            S = E;  
        }else  
            S = S*10 + E;  
    }  
}  
  
void Cola::sacar(int &E) {  
    if(S!=0){  
  
        int x = S/(int)pow(10,cantidadD(S)-1)/1;//(int) para parte entera  
        E = x;  
        S = S%(int)pow(10,cantidadD(S)-1);  
    }  
}
```

Pregunta 3

Finalizado

Puntúa como 1,00

Implemente el metodo `inserta_primer` (dato) perteneciente a la clase `Lista` que tiene como unico atributo `PtrLista` de tipo puntero a `Nodo` con la siguiente estructura (prev,dat,next) , se hace notar que dato es el valor a insertar



```

void Lista::inserta_primer(int e){
    Nodo* x = new Nodo();
    if(x!=NULL){
        x->dat = e;
        x->next = ptrElementos;
        x->prev = fin()
        primero()->prev = x;
        ptrElementos = x;
    }else{
        cout<<"ERROR: No existe espacio en la memoria.";
    }
}

```

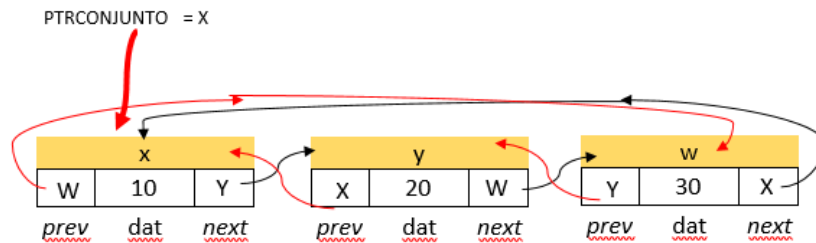
Pregunta 4

Finalizado

Puntúa como 1,00

Implemente el metodo denominado **Ordinal(E)** perteneciente a la clase Conjunto que tiene como unico atributo PtrConjunto de tipo puntero a Nodo (prev,dat,next)

Se hace notar que E es una parametro por valor y que el metodo retorna un valor numerico.



```

int Conjunto::Ordinal(int e) {
    int cont = 0;
    Nodo* aux = PtrConjunto;
    while (aux != NULL) {
        cont++;
        if (cont == e)
            return cont;
        aux = aux->sig;
    }
    return 0;
}

```

Pregunta 5

Finalizado

Puntúa como 1,00

Implemente el metodo denominado EsCompleto perteneciente a la clase Polinomio, que tiene por unico atributo la lista L donde se guardan dos valores por cada termino(el metodo retorna un valor booleano)

Ej. si el polinomio es $2x^3+5x^6$ Entonces la lista L tiene <2,3,5,6>

Polinomio completo: Respecto a una variable, es aquel donde dicha variable presenta todos los exponentes desde 0 hasta el mayor incluso.

..

//LISTA CON VECTORES

```
bool Polinomio::EsCompleto(){
    int c=0;
    for(int i = 1, i<=numeroTerminos();i++){
        int t = buscarTerminoN(i);

        if(siguiente(t)>=0 && siguiente(t)<=grado)
            c++;

    }
    if(c==grado()+1)
        return true;
    return false;
}
```

