



(b) 5 elements

(c) All the names in the alphabetical order:

Adam, Alan, Bill, Bob, Cat, Cindy, Dan, David, Eric, Eva

So the elements in the Next array should be :

0	1	2	3	4	5	6	7	8	9
6	-1	9	4	0	8	2	3	7	1

2. Assume the array index **starts from 0**.

(1)

Structure Stack

```
{
  top;
  array;
}
```

The array index of top in the stack

An array to save elements

InitStack()

```
{
  Stack.top = -1;
}
```

IsEmpty()

```
{
  If Stack.top is -1
    Return true;
  Else
    Return false;
}
```

```
Push(input)
```

The element we want to push into stack.

```
{
  Stack.top = Stack.top + 1;
  Stack.array[ Stack.top ] = input;
}
```

Pop()

```

{
    If IsEmpty()
        Return "Stack is empty";
    Else
    {
        Top = Stack.top;
        Stack.top = Stack.top - 1
        Return Stack.array[ Top ];
    }
}

```

(2)

Structure Queue

```

{
    head;
    tail;
    array;
};

```

```
};
```

InitQueue()

```

{
    Queue.head = -1;
    Queue.tail = -1;
}

```

IsEmpty()

```

{
    If Queue.head is equal to Queue.tail
        Return true;
    Else
        Return false;
}

```

```
}
```

Enqueue(input)

```

{
    Queue.tail = Queue.tail + 1;
    Queue.array[Queue.tail] = input;
}

```

Dequeue()

```

{
    If IsEmpty()
        Return "Queue is empty";
}

```

```

    Else
    {
        Queue.head = Queue.head + 1;
        Return Queue.array[Queue.head]
    }
}

```

(3) Assume size of the array is ArraySize

Structure Queue

```

{
    head;
    tail;
    flag;
    array;
};

```

```

};

```

InitQueue()

```

{
    Queue.head = -1;
    Queue.tail = -1;
    flag = 0;
}

```

IsEmpty()

```

{
    If flag is 0
        Return true;
    Else
        Return false;
}

```

IsFull()

```

{
    If flag is equal to ArraySize
        Return true;
    Else
        Return false;
}

```

Enqueue(input)

```

{
    If IsFull()
        Return "Queue is full";
}

```

```
Else
{
    Queue.tail = ( Queue.tail + 1 )mod ArraySize;
    Queue.array[ Queue.tail ] = input;
    flag = flag + 1;
}
}
Dequeue()
{
    If IsEmpty()
        Return "Queue is empty";
    Else
    {
        flag = flag - 1;
        Queue.head = (Queue.head+1)mod ArraySize;
        Return Queue.array[Queue.head];
    }
}
```

Calculate the remainder