Yash Agarwal

21104039.

Page 1

Q1

```
# include <iostream>
using namespace std;
 Class Node {

     int marks ;
     node * next ;

     node (int m) {
         this -> marks = m ;
         this -> next = Null ;
     }

 void create ( node * & head )
 {

     int m ;
     cout << "enter marks" ;
     cin>>m ;
     node * n = new node (m) ;
     if (head == Null)
     {
         head = n ;
         return ;
     }
     node * temp = head ;
     while (temp -> next != Null)
     {
         temp = temp -> next ;
     }
     temp -> next = m ;

}
```

```
void solve ( node * & head ) {
    Node * i = head ;
    while (i != head) {

        int temp = i -> marks ;
        node * j = i ;
        node * min = j ;

        while (j != Null) {
            if (j -> marks < min -> marks)
                min = j ;
            j = j -> next ;
        }
        i -> marks = min -> marks ;
        min -> marks = temp ;
        i = i -> next ;

    }
    int l = 0
    node * i = head ;
    while ( i != Null) {

        i = i -> next ;
        l += 1 ;

    }
    int count = 0 ;
    i = head ;
    while ( count < l - 3 && i != Null ) {

        count ++ ;
        i = i -> next ;
    }
    if (i != Null) {
        cout << "3 elements are not present" ;
        return ;
    }
cout << "the 3 toppers are" << endl;
  while (i != NULL){
     cout << i->marks;
       i = i->next;
  }
 }
};

int main(){
    Node* head = NULL;
    int n;
    cout << "Enter the length of link list";
    cin>>n;

    for(int i=0; i<n; i++){

          head -> create(head);
    }

    head -> ~~sort~~ solve (head);

}
```

<u>Q2</u>

```
~~2i+1 < n~~

~~void check (~~ ~~int arr[ ],n) }~~
~~{~~

~~for(int i=1; 2i+1<n; i++)~~
~~{~~
~~y(arr[2i] != arr[2i+1])~~
~~{~~
~~cout << "Not identical"~~
~~return;~~
~~{~~
~~}~~
```

Q2

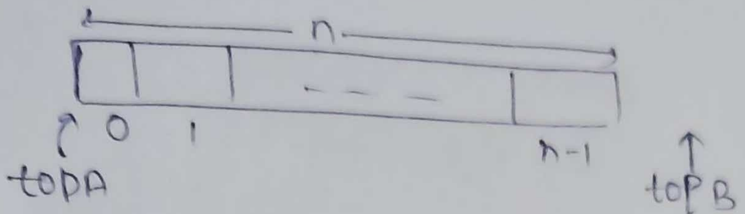
```
~~void check (int arr[ ], n)~~
~~{ if(arr[1] != arr[2]) count~~
~~//if n = 0, root is present.~~
~~for(int i=1;~~
~~if (arr[1] != arr[2])~~
~~cout << "~~

void check (int*arr1, int *arr2,
int m, int n){
      if (m != n){
          cout << "Not identical";
          return;
      }

   for(int i=0; i<n; i++ ){

      if (arr1[i] != arr2[i])
      {
          cout << "Not identical";
          return;
      }
   }
   cout << "identical";
   return;
```

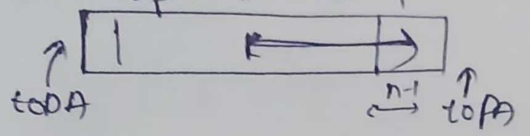
Q 3



top A

Underflow ⇒ top A < 0

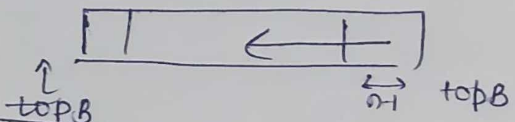
Overflow ⇒ top A > n-1



top B

underflow ⇒ top B > n-1

Overflow ⇒ top B < 0



Q 4

```
Void insertion sort (int *arr, int n, int i, int j,
  int temp) {
    if (i>n) return;
    temp = arr[i];
    [illegible]

    if (j<0) {
        arr[j+1] = temp;
        insertion sort (arr, n, i+1, i, temp);
    }
    if (arr[j] > temp ){
        arr[j+1] = arr [j];
        insertion sort (arr, n, i, j-1, temp); }
    }
    else {
        arr[j+1] = temp;
        insertionsort (arr, n, i+1, j, temp);
} }
```

~~void insertion sort (int * arr, n, i, j, temp)~~
~~{~~
~~if (j<0) return;~~
~~if (i>n) return;~~
~~if (i>n) return;~~
~~[illegible]~~
~~[illegible]~~
~~for(i=1; i<n; i++)~~
~~j=i~~
~~j=i;~~
~~int temp = arr[i];~~
~~if (i>n) return;~~
~~if (j<0){~~
~~arr[j+1] = temp;~~
~~i++;~~
~~return;~~
~~}~~
~~if (arr[j] > temp) {~~
~~arr[j] = arr [j+1];~~
~~}~~
~~else (~~
~~arr[j+1] = temp;~~
~~i++;~~
~~return~~
~~}~~
~~j = i;~~
~~temp = arr [i];~~
~~insertion sort (arr, n, i, j-1; temp);~~