1. Write a function to find the maximum element in the stack.

```
Ans.#include <stdio.h>
#include <stdlib.h>
int data[100000],top,max=0;
void push()
{
   int item;
   scanf("%d",&item);
   top++;
   data[top]=item;
   if(max < data[top])
        max = data[top];
}

void pop()
{
    int i;
    if(max == data[top])
        max=0;
    top--;
    for(i= top;i>=0;i--)
        if(max < data[i])
            max = data[i];
}
int main()
{
    int t,n,choice;
    top = -1;
    scanf("%d",&t);
    while(t--)
    {
        scanf("%d",&choice);
        switch(choice)
        {
            case 1 : push();
                     break;
            case 2:  pop();
                     break;
            case 3: printf("%d\n",max);
                    break;
        }
    }
    return 0;

}
```

2. Write a program to find the minimum element in the stack.

```
Ans.#include <stdio.h>
int main()
{
    int q;
    scanf("%d",&q);
    int stack[q],stackmin[q];
    int top=-1,topmin=-1;
    while(q--)
    {
        int x;scanf("%d",&x);
        if(x==1)
        {
           int y;scanf("%d",&y);
           stack[++top]=y;
           if(topmin==-1)
             stackmin[++topmin]=y;
           else if(y<=stackmin[topmin])
               stackmin[++topmin]=y;
       }
       else if(x==2)
       {
           if(top==-1)
               printf("-1\n");
           else
           {
               if(stack[top]==stackmin[topmin])
                   topmin--;
               //printf("%d\n",stack[top]);
               top--;}
       }
       else if(x==3)
       {
           if(top==-1)
               printf("-1\n");
           else
               printf("%d\n",stack[top]);}
           else
           {
               if(top==-1)
                   printf("-1\n");
               else
                   printf("%d\n",stackmin[topmin]);}
   }
   return 0;
}
```