1.Queue via lists

```
#include <stdio.h>
#include <stdlib.h>
struct Queue {
   int data;
   struct Queue* next;
};
int cap=0;
int max=3;
void insert(struct Queue** head, int value) {
   if (cap<max)
   {
      struct Queue* newNode = (struct Queue*)malloc(sizeof(struct Queue));
      struct Queue* temp = *head;
      newNode->data = value;
      newNode->next = NULL;

      if (*head == NULL) {
         *head = newNode;
         return;
      }

      while (temp->next != NULL) {
         temp = temp->next;
```

```
        }

        temp->next = newNode;
        cap++;
    }
    else{
        printf("Overflow.\n");
    }
}
void delete(struct Queue** head) {
    if (cap<0) {
        printf("Underflow.\n");
        return;
    }
    struct Queue* temp = *head;
    *head = (*head)->next;
    free(temp);
    cap--;
}
void display(struct Queue* head) {
    struct Queue* temp = head;

    if (temp == NULL) {
        printf("Queue is empty.\n");
        return;
    }
```

```
    while (temp != NULL) {
        printf("%d -> ", temp->data);
        temp = temp->next;
    }
    printf("NULL\n");
}

int main() {
    struct Queue* head = NULL;
    int c,a;
    printf("Enter 1 to insert, 2 to delete, 3 to display and 4 to end program.\n");
    scanf("%d",&c);
    while(1)
    {
        switch(c)
        {
            case(1):
            {
                printf("Enter element to insert.\n");
                scanf("%d",&a);
                insert(&head,a);
                break;
            }
            case(2):
            {
```

```
            delete(&head);
            break;
        }
        case(3):
        {
            display(head);
            break;
        }
        case(4):
        {
            exit(1);
            break;
        }
    }
    printf("Enter 1 to insert, 2 to delete, 3 to display and 4 to end program.\n");
    scanf("%d",&c);
  }
}
```

```
Enter element to insert.
4
Enter 1 to insert, 2 to delete, 3 to display and 4 to end program.
3
2 -> 4 -> NULL
Enter 1 to insert, 2 to delete, 3 to display and 4 to end program.
2
Enter 1 to insert, 2 to delete, 3 to display and 4 to end program.
2
Enter 1 to insert, 2 to delete, 3 to display and 4 to end program.
2
Underflow.
Enter 1 to insert, 2 to delete, 3 to display and 4 to end program.
3
Queue is empty.
Enter 1 to insert, 2 to delete, 3 to display and 4 to end program.
```

2. Stack via list

```
#include <stdio.h>
#include <stdlib.h>
struct Stack {
   int data;
   struct Stack* next;
};
int top=-1;
int max=3;
void push(struct Stack** head, int value) {
   if (top<max-1)
   {
```

```
        struct Stack* newNode = (struct Stack*)malloc(sizeof(struct Stack));
        struct Stack* temp = *head;
        newNode->data = value;
        newNode->next = NULL;

        if (*head == NULL) {
            *head = newNode;
            return;
        }

        while (temp->next != NULL) {
            temp = temp->next;
        }

        temp->next = newNode;
        top++;
    }
    else{
        printf("Overflow.\n");
    }
}
void pop(struct Stack** head) {
    if (top==-1) {
        printf("Underflow.\n");
        return;
    }
    struct Stack* temp = *head;
    struct Stack* prev = NULL;
```

```
    while (temp->next != NULL) {
        prev = temp;
        temp = temp->next;
    }
    if (prev == NULL) {
        *head = NULL;
    } else {
        prev->next = NULL;
    }
    free(temp);
    top--;
}
void display(struct Stack* head) {
    struct Stack* temp = head;

    if (temp == NULL) {
        printf("Stack is empty.\n");
        return;
    }

    while (temp != NULL) {
        printf("%d -> ", temp->data);
        temp = temp->next;
    }
    printf("NULL\n");
}
```

```
int main() {
   struct Stack* head = NULL;
   int c,a;
   printf("Enter 1 to insert, 2 to delete, 3 to display and 4 to end program.\n");
   scanf("%d",&c);
   while(1)
   {
      switch(c)
      {
         case(1):
         {
            printf("Enter element to insert.\n");
            scanf("%d",&a);
            push(&head,a);
            break;
         }
         case(2):
         {
            pop(&head);
            break;
         }
         case(3):
         {
            display(head);
            break;
         }
         case(4):
         {
```

```
            exit(1);
            break;
        }
    }
    printf("Enter 1 to insert, 2 to delete, 3 to display and 4 to end program.\n");
    scanf("%d",&c);
  }
}
```

```
PS C:\Users\bmsce\Desktop\DSCS235> cd "c:\Users\bmsce\Desktop\DSCS235\" ; if ($?
rFile }
Enter 1 to insert, 2 to delete, 3 to display and 4 to end program.
1
Enter element to insert.
2
Enter 1 to insert, 2 to delete, 3 to display and 4 to end program.
1
Enter element to insert.
4
Enter 1 to insert, 2 to delete, 3 to display and 4 to end program.
1
Enter element to insert.
6
Enter 1 to insert, 2 to delete, 3 to display and 4 to end program.
3
2 -> 4 -> 6 -> NULL
Enter 1 to insert, 2 to delete, 3 to display and 4 to end program.
2
Enter 1 to insert, 2 to delete, 3 to display and 4 to end program.
3
2 -> 4 -> NULL
Enter 1 to insert, 2 to delete, 3 to display and 4 to end program.
```