```cpp
1   //step 1, add these important funvtions. Priotirty rank is in number. lower
      num is higher
2   #include<stdio.h> //1
3   #include<stdlib.h> //2
4   #include<string.h> //3
5   #include<ctype.h> //4
6   #include<math.h> //5
7   #include<time.h> // 6
8   //so on
9
10
11  #define pauseEnter() while(getchar() != '\n') //defines newer getchar holds
12  #define rinseScreen() {system("cls"); for(int i=0; i < 30; i++)printf("\n");}
13
14  struct datas { //Step 2, create structures
15      int number;
16
17      datas *prev, *next;
18  }*head = NULL, *tail = NULL;
19
20  void freeAll() { //Step 3, create erase all
21      datas *curr;
22      curr = head;
23
24      while (head != NULL) {
25          curr = head;
26          head = head->next;
27          free(curr);
28      }
29  }
30
31  void push(int mode, int numbero) { //Step 5, create adding funtion
32      datas *curr = (datas *)malloc(sizeof(datas)); //Step 5.5, Complete all
          templates
33      curr->next = curr->prev = NULL;
34
35      /* fill content*/
36      curr->number = numbero;
37      /*fill end*/
38
39      if (mode = -1) {//pushHead
40          if (head == NULL) {
41              head = tail = curr;
42          }
43          else { //Head Differ
44              head->prev = curr;
45              curr->next = head;
46              head = curr;
47          }
48      }
49      else if (mode = 0) {//pushBetween
50
51      }
52      else if (mode = 1) {//pushTail
53          if (head == NULL) {
54              head = tail = curr;
```

```cpp
55              }
56          else { //Taill Differ
57              tail->next = curr;
58              curr->prev = tail;
59              tail = curr;
60          }
61      }
62      tail->next = NULL;
63      head->prev = NULL;
64  }
65
66  void pop(int mode) {
67      if (head) {
68          if (head == tail) {
69              free(head);
70              head = tail = NULL;
71          }
72          else {
73              if (mode == -1) { //popHead
74                  head = head->next;
75                  free(head->prev);
76                  head->prev = NULL;
77              }
78              else if (mode == 0) { //popBetween
79
80              }
81              else if (mode == 1) { //popTail
82                  tail = tail->prev;
83                  free(tail->next);
84                  tail->next = NULL;
85              }
86          }
87      }
88      else {
89          printf("no data!\n");
90      }
91  }
92
93  void printDatas() { //Step 7, printf is worthless!
94      datas *curr = head;
95
96      printf("NULL<->");
97      while (curr != NULL) {
98          printf("%d<->", curr->number);
99          curr = curr->next;
100     }
101     printf("NULL\n");
102 }
103
104 void inits() { //Step 4, put in init or main
105     freeAll();
106     rinseScreen();
107 }
108
109 int main() {
110     inits(); //Step 4.5, here is init!
```

```
111
112        push(1, 5); //Step 8, add fun spice
113        push(1, 6);
114        push(-1, 10);
115        push(-1, 11);
116
117        printDatas();
118
119        pop(1);
120        printDatas();
121        pop(-1);
122        printDatas();
123
124        pauseEnter();
125        return 0; //Step 9, do the rest of the jobs!
126    }
127
128    //Step 10, enjoy!!!
```