```cpp
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<ctype.h>
#include<math.h>
#include<time.h>

void rinseScreen(){
    system("cls");
    for(int i = 0; i < 25; i++){
        printf("\n");
    }
}

struct Data{
    char Name[31];
    int quantity;

    Data *next;
} *head = NULL, *tail = NULL;

void pushHead(char Name[], int quantity){

    Data *node = (Data *) malloc(sizeof(Data));

    //(*node).quantity = quantity;
    node->quantity = quantity;
    strcpy(node->Name,Name);

    if(head==NULL){
        head = tail = node;
    } else{
        node->next = head;
        head = node;
    }
    tail->next = NULL;
}

void pushTail(char Name[], int quantity){

    Data *node = (Data *) malloc(sizeof(Data));

    node->quantity = quantity;
    strcpy(node->Name,Name);

    if(head==NULL){
        head = tail = node;
    } else{
        tail->next = node;
        tail = node;
    }
    tail->next = NULL;
}

void push(char Name[], int a){
    if(head==NULL){
```

```
57          pushHead(Name, a);
58      }else{
59          if(a < head->quantity){
60              pushHead(Name, a);
61              //if quantity < head
62          }else if (a > tail->quantity){
63              pushTail(Name, a);
64              //if quantity > head
65          }else{
66              //if head < quantity < tail (between)
67              Data *temp = head;
68              while(temp->next->quantity < a){
69                  temp = temp->next;
70              }
71              Data *node = (Data *) malloc(sizeof(Data));
72              node->quantity = a;
73              strcpy(node->Name,Name);
74              node->next = temp->next;
75              temp->next = node;
76          }
77      }
78  }
79
80  void pop(int a){
81      //check if there is data on the list
82      if(head){//if there is data
83          if(head->quantity == a){
84              //if quantity to delete is on head
85              if(head==tail){//if data is left 1 on the list
86                  free(head);
87                  head = tail = NULL;
88              } else{//if data > 1 on the list
89                  Data *temp = head;
90                  head = head->next;
91                  free(temp);
92              }
93          } else if(tail->quantity == a) { //if quantity to delete is on tail
94              Data *temp = head;
95              while(temp->next != tail){
96                  temp = temp->next;
97              }
98              free(tail);
99              tail = temp;
100             tail->next = NULL;
101         } else {
102             //if quantity to delete is in-between, middle
103             Data *temp = head;
104             while(temp != NULL && temp -> quantity != a){
105                 temp = temp ->next;
106             }
107             if(temp!=NULL){//if data being search is found
108                 Data *temp2 = head;
109                 while(temp2->next != temp){
110                     temp2= temp2 ->next;
111                 }
112                 temp2->next = temp->next;
```

```
113                 free(temp);
114 
115             }
116         }
117     }else{//if data not on the list
118         printf("Data Kosong");
119     }
120 
121 }
122 
123 void popName(char Nama[]){
124     //check if there is data on the list
125     if(head){//if there is data
126         if(strcmp(head->Name, Nama) == 0){
127             //if quantity to delete is on head
128             if(head==tail){//if data is left 1 on the list
129                 free(head);
130                 head = tail = NULL;
131             } else{//if data > 1 on the list
132                 Data *temp = head;
133                 head = head->next;
134                 free(temp);
135             }
136         } else if(strcmp(tail->Name,Nama) == 0) { //if quantity to delete is
            on tail
137             Data *temp = head;
138             while(temp->next != tail){
139                 temp = temp->next;
140             }
141             free(tail);
142             tail = temp;
143             tail->next = NULL;
144         } else {
145             //if quantity to delete is in-between, middle
146             Data *temp = head;
147             while(temp != NULL && strcmp(temp -> Name, Nama) != 0){
148                 temp = temp ->next;
149             }
150             if(temp!=NULL){//if data being search is found
151                 Data *temp2 = head;
152                 while(temp2->next != temp){
153                     temp2= temp2 ->next;
154                 }
155                 temp2->next = temp->next;
156                 free(temp);
157 
158             }
159         }
160     }else{//if data not on the list
161         printf("Data Empty");
162     }
163 
164 }
165 
166 void popIDbased(int a){
167     //char Target[31];
```

```
168     Data *take = (Data *) malloc(sizeof(Data *));
169     take = head;
170     for(int i = 0; i < a; i++){
171         //printf("Choose = %s\n", take->Name);
172         take = take->next;
173     }
174     //strcpy(Target, take->Name);
175     popName(take->Name);
176 }
177 
178 void viewList(){
179     Data *temp;
180     temp = head;
181     while(temp!=NULL){
182         printf("%d->", temp->quantity);
183         temp = temp->next;
184     }
185     printf("NULL");
186 }
187 
188 void PrintBits(int a){
189     Data *temp;
190     temp = head;
191 }
192 
193 void PrintAll(){
194     int scounte = 0;
195     Data *temp;
196     temp = head;
197     scounte++;
198     printf("                    --- ORDER LIST ---\n");
199     printf("\n");
200     printf(" -------+--------------------------------+-----------\n");
201     printf(" | %-4s | %-30s | %-8s |\n", "No.", "Name of Parts", "Quantity");
202     printf(" -------+--------------------------------+-----------\n");
203     while(temp !=NULL){
204         printf(" | %-4d | %-30s | %-8d |\n", scounte, temp->Name, temp-
              >quantity);
205         temp= temp->next;
206         scounte++;
207     }
208     printf(" -------+--------------------------------+-----------\n");
209 }
210 
211 int main(){
212     char* fmt = "%[^\n]%*c";
213     int select=0, kounter=0;
214     char InsertName[31];
215     int InsertQuantity=0;
216     int InsertID=0;
217 
218     do{
219         rinseScreen();
220 
221         printf(" BLUE MOTORCYCLE PARTS\n");
222         printf(" ......................\n");
```

```
223         printf("\n");
224         printf(" 1. View Order List\n");
225         printf(" 2. Add New Order\n");
226         printf(" 3. Take Order\n");
227         printf(" 4. Exit\n");
228         printf("\n");
229         printf(" >> Input choice : ");
230 
231         scanf("%d", &select); fflush(stdin);
232 
233         switch(select){
234         case 1: //view order list
235             rinseScreen();
236             PrintAll();
237             getchar(); getchar(); //pls help, Visual Studio 2017 is broken,
                  skips getchar();
238             break;
239         case 2: //add new order
240             //printf("\n");
241             //getchar();
242             do{
243                 printf("\n Input Name of Motorcycle's Part [3..30]: ");
244                 getchar();
245                 scanf(fmt, InsertName); fflush(stdin);
246             } while (strlen(InsertName) > 30 || strlen(InsertName) < 3);
247 
248             printf("\n");
249 
250             do{
251                 printf(" Input Quantity of The Motorcycle's Part [1..20]: ");
252                 scanf("%d", &InsertQuantity); fflush(stdin);
253             }while (InsertQuantity > 20 || InsertQuantity < 1);
254 
255             printf("\n");
256 
257             pushHead(InsertName, InsertQuantity);
258             kounter++;
259 
260             printf("\n");
261 
262             printf("  --- Add New Order Success --- ");
263 
264             getchar(); getchar();
265             break;
266         case 3: //take order
267             if(head){
268                 rinseScreen();
269                 PrintAll();
270 
271                 printf("\n\n");
272 
273                 do{
274                 printf(" Input Number of The Order [1..%d]: ", kounter);
275                 scanf("%d", &InsertID); fflush(stdin);
276                 } while(InsertID > kounter || InsertID < 1);
277 
```

```
278                 printf("\n");
279 
280                 //func
281                 popIDbased(InsertID-1);
282                 kounter--;
283 
284                 printf("\n");
285 
286                 printf(" --- Take Order Success --- ");
287 
288             } else {
289                 printf("\n\n");
290                 printf("  --- There is No Order in The List --- ");
291             }
292 
293             getchar(); getchar();
294             break;
295         case 4: //exit
296             break;
297         default:
298             break;
299         }
300 
301     } while (select != 4);
302 
303     return 0;
304 }
```