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

//Codeblock version! Visual Studio 2017 is broken!

/* Differences:
Queue = (FIFO) First in First out = pushHead(), popTail();
Stack = (LIFO) Last in First out = pushHead(), popHead();

*/

void PauseEnter(){
    printf("\n[ENTER] Continue");
    getchar(); //getchar(); //Visual Studio 2017 is broken, igonred first getchar! Please use Eclipse instead next time!
}

void RinseScreen(){
    system("cls"); //unprofessional! but Visual Studio has no clrscr(); :()
    for(int i = 0; i < 25; i++){
        printf("\n");
    }
}

struct Data{ //Queue it
    char Customer[100];
    char Dress[100];
    float Price;

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


void pushHead(char Customer[], char Dress[], float price){
    nd = (Data *) malloc(sizeof(Data));

    strcpy(nd->Customer, Customer);
    strcpy(nd->Dress, Dress);
    nd->Price = (float) price;


    if(head==NULL){
        head=tail=nd;
    } else {
        head->prev = nd; //Forward arrow
        nd->next = head; //Backward arrow, remember to connect back to it's predecessor
        head = nd;
    }
    tail->next = NULL;
    head->prev = NULL;
}

void pushTail(char Customer[], char Dress[], float price){
    nd = (Data *) malloc(sizeof(Data));

    strcpy(nd->Customer, Customer);
    strcpy(nd->Dress, Dress);
    nd->Price = (float) price;

    if(head==NULL){
        head=tail=nd;
    } else {
        tail->next = nd; //Forward arrow
        nd->prev = tail; //Backward arrow, remember to connect back to it's predecessor
        tail = nd;
```

```
71      }
72      tail->next = NULL;
73      head->prev = NULL;
74  }
75
76  void popHead(){ //erase first data from head
77      if(head){ //head != NULL
78          //if there is data
79          if(head==tail){
80              //if data is only consists 1 set
81              free(head); //or tail, whatever.
82              head = tail = NULL;
83          } else {
84              head = head->next;
85              free(head->prev); //big advantage of double linked list, no need nd.
                reduces data loss
86              head->prev = NULL; //reassign deleted data as NULL
87          }
88      } else {
89          //if there is no data
90          printf("tidak ada data");
91      }
92  }
93
94  void popTail(){ //erase first data from tail
95      if(head){ //head != NULL
96          //if there is data
97          if(tail==head){
98              //if data is only consists 1 set
99              free(tail); //or head, whatever.
100             head = tail = NULL;
101         } else {
102             tail = tail->prev;
103             free(tail->next); //big advantage of double linked list, no need nd.
                reduces data loss
104             tail->next = NULL; //reassign deleted data as NULL
105         }
106     } else {
107         //if there is no data
108         printf("\n======\nNo Data!\n======\n");
109     }
110 }
111
112 void popSearch(char name[]){ //this one is better
113     if(head){
114         nd = head;
115
116         //nd->weight != (float) weight
117         while(nd!=NULL && strcmp(nd->Customer, name) != 0){
118             nd = nd->next;
119         }
120         if(nd==NULL){
121             printf("Name not found");
122         } else {
123             if(nd==head)popHead();
124             else if(nd==tail)popTail();
125             else {
126                 nd->prev->next = nd->next;
127                 nd->next->prev = nd->prev;
128                 free(nd);
129             }
130         }
131     }
132 }
133
134 void popAll(){
135     while(head != NULL) {
136         popHead();
137     }
138 }
139
140 void view(){
141     nd = head;
```

```
142         printf("NULL<->");
143         while(nd != NULL){
144             printf("%f<->", nd->Price);
145             nd = nd->next;
146         }
147         printf("NULL");
148     }
149 
150     void printSeparator(int Extender[], bool withEndLine){
151         for(int i = 0; i < Extender[0]; i++) printf("-");
152         printf("+");
153         for(int i = 0; i < Extender[1]; i++) printf("-");
154         printf("+");
155         for(int i = 0; i < Extender[2]; i++) printf("-");
156         printf("+");
157         for(int i = 0; i < Extender[3]; i++) printf("-");
158         if(withEndLine == true){
159             printf("\n");
160         } else {
161             printf("");
162         }
163 
164     };
165 
166     void printDatas(int number, int maximal){
167         RinseScreen();
168         nd = head;
169         //static int Kountmer = 0;
170         int ListKount = 1;
171         static int FunnyKount = 0;
172         //Kountmer = number;
173         if(number > 0) FunnyKount = 0;
174 
175         int Extender[4]; char TableHead[4][100];
176 
177         strcpy(TableHead[0], "No.");
178         strcpy(TableHead[1], "Customer's Name");
179         strcpy(TableHead[2], "Dress Name");
180         strcpy(TableHead[3], "Price");
181 
182         Extender[0] = strlen(TableHead[0]) + 3;
183         Extender[1] = strlen(TableHead[1]) + 7;
184         Extender[2] = strlen(TableHead[2]) + 7;
185         Extender[3] = strlen(TableHead[3]) + 5;
186 
187         if(number <= maximal-10 ) printf("         --- DESOLATE QUEUE VIEW ---  \n");
188         else if(number >= maximal-9 && number <= maximal-4 ) printf("         --- QUEUE
        VIEW ---  \n");
189         else if(number >= maximal-3 ) printf("         --- CROWDED QUEUE VIEW ---  \n");
190 
191         printf(" \n");
192 
193         printSeparator(Extender, true);
194 
195         printf("| %-*s | %-*s | %-*s | %-*s |", Extender[0]-3, TableHead[0],
        Extender[1]-2, TableHead[1], Extender[2]-2, TableHead[2], Extender[3]-3,
        TableHead[3]);
196         printf("\n");
197 
198         printSeparator(Extender, false);
199 
200         if(head){
201             FunnyKount = 0;
202             while(nd != NULL){
203                 printf("\n| %*d. | %-*s | %-*s | $ %*.0f |", Extender[0]-4, ListKount,
                Extender[1]-2, nd->Customer, Extender[2]-2, nd->Dress, Extender[3]-5,
                (float) nd->Price);
204                 ListKount++;
205                 //Kountmer++;
206                 nd = nd->next;
207             }
208         } else {
209             printf("\n --- There is no customer in queue --- \n");
```

```
210             //printf("\n%d\n", FunnyKount);
211             if(FunnyKount >= 3 && FunnyKount <= 7) printf("\n --- Ya udah bro/sis -_-
                sepi ya sepi! ngapain juga dibahas? Sana tuh, cari customer! --- \n");
212             else if(FunnyKount >= 8 && FunnyKount <= 11) printf("\n --- Ampun dah
                bro/sis. dibilangin sepi masih ngeyel! Sono, Pilih [Add Customer] biar gk
                sepi! --- \n");
213             else if(FunnyKount >= 12 && FunnyKount <= 15) printf("\n --- Hadeuh! cape
                deeehhh... --- \n");
214             else if(FunnyKount >= 16 && FunnyKount <= 17) printf("\n --- Ah males ah!
                gak mau bantu! --- \n");
215             FunnyKount++;
216         }
217
218         printf("\n");
219         printSeparator(Extender, true);
220
221     }
222
223
224     void menus(int kountings, int maximal, int & tutoriala, int & tutoriali){
225         RinseScreen();
226         //static int tutoriala = 0;
227         //static int tutoriali = 0;
228
229         printf("BLUE DRESS SHOP QUEUE\n");
230         printf("####################\n");
231         printf("\n");
232         printf("1. View Queue (%d queues)\n", kountings);
233         printf("2. Add Customer to Queue ");
234
235         if(kountings == 0){
236             if(tutoriala == 0){
237                 printf("(<- Select this to add 1st and more customer!)");
238                 tutoriala++;
239             } else {
240                 printf("(<-)");
241             }
242         }
243         printf("\n");
244
245         printf("3. Serve Customer ");
246
247         if(kountings > 0){
248             if(tutoriali == 0){
249                 printf("(<- Select this to serve your customer! It will serve the
                    earliest one."); //)
250                 tutoriali++;
251             } else printf("(<- ");
252         }
253
254         if(kountings >= maximal-4 && kountings < maximal-1) printf("Please serve your
            customer now! Max %d customers!", maximal);
255         else if(kountings >= maximal) printf("PLEASE SERVER YOUR CUSTOMER RIGHT NOW!!!
            MAX %d IS REACHED!", maximal);
256         if(kountings > 0) if(tutoriali == 1) printf(")");
257         printf("\n");
258
259         printf("4. Exit\n");
260         printf("\n");
261         printf(">>Input Choice : ");
262
263     }
264
265     void addCustomer(int & kountings, int maximal);
266     void serveCustomer(int & kountings, int maximal);
267
268     int main(){
269         int select;
270         int kounter = 0;
271         int maximal = 10;
272
273         int sutorials[2] = {0,0};
274
```

```
275         do{
276             menus(kounter, maximal, sutorials[0], sutorials[1]);
277             scanf("%d", &select);
278             fflush(stdin);
279 
280             switch(select){
281             case 1: //view queue
282                 printDatas(kounter, maximal);
283                 PauseEnter();
284                 break;
285             case 2: //add customer to queue
286                 addCustomer(kounter, maximal);
287                 break;
288             case 3: //serve customer
289                 serveCustomer(kounter, maximal);
290                 break;
291             case 4: //exit
292                 break;
293             default:
294                 break;
295             };
296         } while (select != 4);
297 
298         //getchar();
299         return 0;
300     }
301 
302     void addCustomer(int & kountings, int maximal){
303         RinseScreen();
304         printf(" --- Add Customer --- \n");
305         printf("######################\n");
306         printf("\n");
307         printf("We have %d queues.\n", kountings);
308         printf("Maximum queue = %d ", maximal);
309         if(kountings >= 10) printf("[REACHED!!!]");
310         printf("\n\n");
311 
312         //CodeBlock with gcc supports string data type. however, because you guys wants
            for Microsoft Visual C++,
313         //No way than to use traditional way.
314         char InsertCharS[2][50];
315         float InsertFloat;
316         //char *numered = "0123456789";
317         bool hasNumber = false;
318         int flagNumberInString = 0; //has non-Alphabet
319 
320         if(kountings < maximal) {
321             do{
322                 //
323                 printf("\nInput Customer's Name [3..20][Must be alphabets (ASCII
                    based)(a-z, A-Z)]: ");
324                 scanf("%49[^\n]", &InsertCharS[0]); fflush(stdin);
325                 for(int i = 0; i < strlen(InsertCharS[0]); i++){
326                     if(
327                         (
328                          (33 <= InsertCharS[0][i] && 64 >= InsertCharS[0][i]) || (91 <=
                             InsertCharS[0][i] && 96 >= InsertCharS[0][i])
329                          ) ||
330                         (
331                          (123 <= InsertCharS[0][i] && 127 >= InsertCharS[0][i]) || (0 <=
                             InsertCharS[0][i] && 31 >= InsertCharS[0][i])
332                          )
333                         ){
334                          flagNumberInString++;
335                     } //Rohit Kumar, Quora; ASCIItable.com. ASCII numbers
336                 }
337                 if(flagNumberInString==0){
338                         hasNumber = false;
339                         //printf("\nNo number\n");
340                 } else {
341                     hasNumber = true;
342                     //printf("\nHas number\n");
343                 }
```

```
344             flagNumberInString = 0;
345         }while((strlen(InsertCharS[0]) < 3 || strlen(InsertCharS[0]) > 20) ||
            hasNumber == true);
346
347         hasNumber = false;
348         flagNumberInString = 0;
349
350         do{
351             printf("\nInput Dress Name [3..20]: ");
352             scanf("%49[^\n]", &InsertCharS[1]); fflush(stdin);
353         }while(strlen(InsertCharS[1]) < 3 || strlen(InsertCharS[1]) > 20);
354
355         do{
356             printf("\nInput Dress Price [$50..$999]: ");
357             scanf("%f", &InsertFloat); fflush(stdin);
358         }while(InsertFloat < (float) 50 || InsertFloat > (float) 999);
359
360         printf("\n| %s | %s | $ %.0f |\n", InsertCharS[0], InsertCharS[1],
            InsertFloat);
361         pushHead(InsertCharS[0], InsertCharS[1], InsertFloat);
362
363         kountings++;
364
365         printf("\n --- Success to Add Customer to Queue List! ---\n");
366         if(kountings >= maximal){
367             printf("\n --- Queue has been full! %d customers now. Please serve them
                now! --- \n", kountings);
368         } else if (kountings >= maximal - 2 && kountings <= maximal - 1) {
369             printf("\n --- Queue is going to full! %d customers now. Please serve
                them now! --- \n", kountings);
370         }
371     } else {
372         printf("\n--- Sorry, Maximum Customer in a Queue is %d ---\n", maximal);
373         printf("\n--- (to employee) Please serve customer before him/her! ---\n");
374     }
375
376     PauseEnter();
377 }
378
379 void serveCustomer(int & kountings, int maximal){
380     RinseScreen();
381     printf(" --- Serve Customer --- \n");
382     printf("#########################\n");
383     char beingServed[50];
384
385     if(head){ //poptail
386         if(tail==head){
387             //if data is only consists 1 set
388             printf("Serving Last customer:\n");
389             printf("\n| %s | %s | $ %.0f |\n", tail->Customer, tail->Dress,
                tail->Price);
390         } else {
391             printf("Serving earlier customer:\n");
392             printf("\n| %s | %s | $ %.0f |\n", tail->Customer, tail->Dress,
                tail->Price);
393         }
394         strcpy(beingServed, tail->Customer);
395         popTail();
396
397         kountings--;
398
399         printf("\n --- %s Has Been Served --- \n", beingServed);
400     } else {
401         printf("\n --- There is No Customer in Queue, Ahhh, relax!!! --- \n");
402     }
403
404     PauseEnter();
405 }
406
407 /*Error list graveyard
408 ((
409                     (
410                      (InsertCharS[0][i] == '0' || InsertCharS[0][i] == '1') ||
```

```
                      (InsertCharS[0][i] == '2' || InsertCharS[0][i] == '3')
                      ) ||
                     (
                      (InsertCharS[0][i] == '4' || InsertCharS[0][i] == '5') ||
                      (InsertCharS[0][i] == '6' || InsertCharS[0][i] == '7')
                      )
                     ) ||
                    (
                     (
                      (InsertCharS[0][i] == '8' || InsertCharS[0][i] == '9') ||
                      (InsertCharS[0][i] == '!' || InsertCharS[0][i] == '@')
                      ) ||
                     (
                      (InsertCharS[0][i] == '#' || InsertCharS[0][i] == '$') ||
                      (InsertCharS[0][i] == '%' || InsertCharS[0][i] == '^')
                      )
                     )) ||
                    (
                     ((
                     (
                      (InsertCharS[0][i] == '&' || InsertCharS[0][i] == '*') ||
                      (InsertCharS[0][i] == '(' || InsertCharS[0][i] == ')')
                      ) ||
                     (
                      (InsertCharS[0][i] == '-' || InsertCharS[0][i] == '_') ||
                      (InsertCharS[0][i] == '+' || InsertCharS[0][i] == '=')
                      )
                     ) ||
                    (
                     (
                      (InsertCharS[0][i] == '`' || InsertCharS[0][i] == '~') ||
                      (InsertCharS[0][i] == '{' || InsertCharS[0][i] == '}')
                      ) ||
                     (
                      (InsertCharS[0][i] == '[' || InsertCharS[0][i] == ']') ||
                      (InsertCharS[0][i] == '|' || InsertCharS[0][i] == '\\')

                     ) //Forget Code (forgetcode.com)
(((InsertCharS[0][i]=='0' || InsertCharS[0][i]=='1') || (InsertCharS[0][i]=='2' ||
InsertCharS[0][i]=='4')) || InsertCharS[0][i]=='5') || ((InsertCharS[0][i]=='6' ||
InsertCharS[0][i]=='7') || (InsertCharS[0][i]=='8' || InsertCharS[0][i]=='9'))
*/

```