

```

1
2
3
4 #include<stdio.h>
5 #include<stdlib.h>
6 #include<string.h>
7 struct car_rent
8 {
9     char car_types[20];
10    int price;
11    struct car_rent *link;
12 };
13 struct car_rent *create(struct car_rent *start);
14 void display (struct car_rent *start);
15 struct car_rent *insert(struct car_rent *start);
16 struct car_rent *del(struct car_rent *start,char car_name[]);
17 struct car_rent *sort (struct car_rent *start);
18 void *search(struct car_rent *start,char car_name[]);
19 struct car_rent *modify(struct car_rent *start,char car_name[]);
20 struct car_rent *front=NULL;
21 struct car_rent *rear=NULL;
22 void create_queue();
23 void del_queue();
24 void display_queue();
25
26 int i,n;
27 void main()
28 {
29     char car_name[20];
30     int ch;
31     struct car_rent *start=NULL;
32     while(1)
33     {
34         printf("\nEnter choice-\n 1) to create\n 2) to create using queue\n 3) to display\n 4)
35         to display using queue\n 5) to insert\n 6) to delete\n 7) to delete using queue\n 8) to
36         sort by price\n 9) to search\n 10) to modify:");
37         scanf("%d",&ch);
38         switch (ch)
39         {
40             case 1:
41                 start=create(start);
42                 break;
43             case 2:
44                 create_queue();
45                 break;
46             case 3:
47                 display(start);
48                 break;
49             case 4:
50                 display_queue();
51                 break;
52             case 5:
53                 start=insert(start);
54                 break;
55             case 6:
56                 printf("Enter car_name you want to delete:");
57                 scanf("%s",car_name);
58                 start=del(start,car_name);
59                 break;
60             case 7:
61                 del_queue();
62                 break;
63             case 8:
64                 sort(start);
65                 break;
66             case 9:
67                 printf("Enter car_name you want to search:");
68                 scanf("%s",car_name);
69                 search(start,car_name);
70                 break;
71             case 10:
72                 printf("Enter car_name you want to modify:");
73                 scanf("%s",car_name);
74                 start=modify(start,car_name);
75                 break;
76             default:
77                 printf("error!");
78         }
79     }
80 }
81
82 struct car_rent *create(struct car_rent* start)
83 {
84     struct car_rent *p,*temp;
85     printf("Enter no of type of car_names:");
86     scanf("%d",&n);

```

```

83     temp=start;
84     for(i=0;i<n;i++)
85     { temp=(struct car_rent*)malloc(sizeof(struct car_rent)*1);
86       printf("enter car_types name:");
87       scanf("%s", (temp->car_types));
88       printf("enter price:");
89       scanf("%d",&(temp->price));
90
91
92       if(start==NULL)
93       {
94         start=temp;
95         temp->link=NULL;
96
97       }
98
99     else
100    {
101      p=start;
102      while(p->link!=NULL)
103      p=p->link;
104      p->link=temp;
105      temp->link=NULL;
106    }
107  }
108  return start;
109  }
110  void display (struct car_rent *start)
111  {
112    struct car_rent *p;
113    if(start==NULL)
114    {
115      printf("\n*Order List is Empty*\n\n");
116      return;
117    }
118    p=start;
119    printf("\ncar_name\tPrice\n");
120    while(p!=NULL)
121    {
122      printf("\n%s\t %d\n",p->car_types,p->price);
123      p=p->link;
124    }
125    printf("\n");
126  }
127
128  struct car_rent *insert(struct car_rent *start)
129  {
130    int pos;
131    struct car_rent *p,*temp;
132    temp=(struct car_rent*)malloc(sizeof(struct car_rent)*1);
133    printf("Enter location you want to insert:");
134    scanf("%d",&pos);
135    if(pos==1)
136    {
137      printf("enter car_types name:");
138      scanf("%s", (temp->car_types));
139      printf("enter price:");
140      scanf("%d",&(temp->price));
141      temp->link=start;
142      start=temp;
143      return start;
144    }
145    p=start;
146    for(i=1;i<pos-1 && p!=NULL;i++)
147    p=p->link;
148    if(p==NULL)
149      printf("\n*less no of car_names*\n\n");
150
151    else
152    {
153      printf("enter car_types name:");
154      scanf("%s", (temp->car_types));
155      printf("enter price:");
156      scanf("%d",&(temp->price));
157      temp->link=p->link;
158      p->link=temp;
159    }
160    return start;
161  }
162
163  struct car_rent *del(struct car_rent *start, char car_name[])
164  {
165    struct car_rent *temp,*p;
166    if(strcmp(start->car_types, car_name)==0)

```

```

167     {
168         temp=start;
169         start=temp->link;
170         free(temp);
171         return start;
172     }
173     p=start;
174     while(p->link!=NULL)
175     { if(strcmp(p->link->car_types,car_name)==0)
176         {
177             temp=p->link;
178             p->link=temp->link;
179             free(temp);
180             return start;
181         }
182         p=p->link;
183     }
184 }
185 struct car_rent *sort (struct car_rent *start)
186 { struct car_rent *p,*q,*temp;
187   temp=(struct car_rent*)malloc(sizeof(struct car_rent)*1);
188   for(p=start;p->link!=NULL;p=p->link)
189   {
190       for(q=p->link;q!=NULL;q=q->link)
191       {
192           if(p->price > q->price)
193           {
194               temp->price=p->price;
195               p->price=q->price;
196               q->price=temp->price;
197               strcpy(temp->car_types,p->car_types);
198               strcpy(p->car_types,q->car_types);
199               strcpy(q->car_types,temp->car_types);
200           }
201       }
202   }
203 }
204 }
205 void *search(struct car_rent *start,char car_name[])
206 {
207     struct car_rent *p;
208     int flag=0;
209     if(strcmp(start->car_types,car_name)==0)
210     {
211         printf("\nCAR_NAME FOUND\n");
212         printf("\ncar_name\tPrice");
213         printf("\n%s\t %d\n",start->car_types,start->price);
214         flag++;
215     }
216     p=start;
217     while(p->link!=NULL)
218     {
219         if(strcmp(p->link->car_types,car_name)==0)
220         {
221             printf("\nCAR_NAME FOUND\n");
222             printf("\ncar_name\tPrice");
223             printf("\n%s\t %d\n",p->link->car_types,p->link->price);
224         }
225         p=p->link;
226         flag++;
227     }
228     if(flag==0)
229         printf("\n\nnot found\n\n");
230 }
231 struct car_rent *modify(struct car_rent *start,char car_name[])
232 {
233     struct car_rent *p;
234     int choice;
235     char ans1,ans2;
236     int flag=0,chw=0;
237     if(strcmp(start->car_types,car_name)==0)
238     {
239         printf("\nCAR_NAME FOUND\n");
240         printf("\ncar_name\tPrice");
241         printf("\n%s\t %d\n",start->car_types,start->price);
242     do
243     {
244         printf("enter:\n 1) to modify car_types name\n 2) to modify car_types price:");
245         scanf("%d",&choice);
246         switch (choice)
247         {
248             case 1:
249                 printf("enter new car_types name:");
250                 scanf("%s",(start->car_types));

```

```

251         break;
252     case 2:
253         printf("enter new car_types price:");
254         scanf("%d",&(start->price));
255         break;
256
257     }
258
259     printf("press 1 to continue modifying:");
260     scanf("%d",&chw);
261     }while(chw==1);
262
263     flag++;
264     return start;
265 }
266
267 p=start;
268 while(p->link!=NULL)
269 {
270     if(strcmp(p->link->car_types,car_name)==0)
271     {
272         printf("\nCAR_NAME FOUND\n");
273         printf("\ncar_name\tPrice");
274         printf("\n%s\t %d\n",p->link->car_types,p->link->price);
275         do
276         {
277             printf("enter:\n 1) to modify car_types name\n 2) to modify car_types price:");
278             scanf("%d",&choice);
279
280             switch (choice)
281             {
282                 case 1:
283                     printf("enter new car_types name:");
284                     scanf("%s", (p->link->car_types));
285                     break;
286                 case 2:
287                     printf("enter new wire_types price:");
288                     scanf("%d",&(p->link->price));
289                     break;
290
291             }printf("press 1 to continue modifying:");
292             scanf("%d",&chw);
293             }while(chw==1);
294
295         }
296         p=p->link;
297         flag++;
298         return start;
299     }
300 }
301
302 if(flag==0)
303     printf("\n\nnot found\n\n");
304 }
305
306 void create_queue()
307 {
308     struct car_rent *temp,*p;
309     int n;
310     printf("Enter Number of types of orders:");
311     scanf("%d",&n);
312     for(int i=0;i<n;i++)
313     {
314
315         if (rear==NULL)
316         {
317             rear=(struct car_rent*)malloc(sizeof(struct car_rent));
318             printf("Enter car_types Name:");
319             scanf("%s", (rear->car_types));
320             printf("Enter car_types Price:");
321             scanf("%d",&(rear->price));
322             rear->link=NULL;
323             front=rear;
324
325         }
326
327         else
328         {
329
330             temp=(struct car_rent*)malloc(sizeof(struct car_rent));
331             printf("Enter car_types Name:");
332             scanf("%s", (temp->car_types));
333             printf("Enter car_types Price:");
334

```

```

335         scanf("%d",&(temp->price));
336         rear->link=temp;
337         rear=temp;
338     }
339 }
340 }
341
342 void display_queue()
343 {
344     struct car_rent *p;
345
346     if(front==NULL)
347     {
348         printf("\n*Queue Underflow*\n");
349     }
350     else
351     {
352         p=front;
353         printf("\ncar_name\tPrice\n\n");
354         while(p!=NULL)
355         {
356             printf("%s\t%d",p->car_types,p->price);
357             printf("\n");
358             p=p->link;
359         }
360     }
361 void del_queue()
362 {
363     struct elec *temp;
364     if(front==NULL)
365     {
366         printf("\n*Queue Underflow*\n");
367     }
368     else
369     {
370         temp=front;
371         front=front->link;
372         free (temp);
373     }
374 }
375
376 /*
377 OUTPUT
378 Enter choice-
379 1) to create
380 2) to create using queue
381 3) to display
382 4) to display using queue
383 5) to insert
384 6) to delete
385 7) to delete using queue
386 8) to sort by price
387 9) to search
388 10) to modify:1
389 Enter no of type of car_names:1
390 enter car_types name:we
391 enter price:223
392
393 Enter choice-
394 1) to create
395 2) to create using queue
396 3) to display
397 4) to display using queue
398 5) to insert
399 6) to delete
400 7) to delete using queue
401 8) to sort by price
402 9) to search
403 10) to modify:2
404 Enter Number of types of orders:1
405 Enter car_types Name:er
406 Enter car_types Price:444
407
408 Enter choice-
409 1) to create
410 2) to create using queue
411 3) to display
412 4) to display using queue
413 5) to insert
414 6) to delete
415 7) to delete using queue
416 8) to sort by price
417 9) to search
418 10) to modify:3

```

```
419
420 car_name      Price
421
422 we           223
423
424
425 Enter choice-
426 1) to create
427 2) to create using queue
428 3) to display
429 4) to display using queue
430 5) to insert
431 6) to delete
432 7) to delete using queue
433 8) to sort by price
434 9) to search
435 10) to modify:4
436
437 car_name      Price
438
439 er           444
440
441 Enter choice-
442 1) to create
443 2) to create using queue
444 3) to display
445 4) to display using queue
446 5) to insert
447 6) to delete
448 7) to delete using queue
449 8) to sort by price
450 9) to search
451 10) to modify:5
452 Enter location you want to insert:1
453 enter car_types name:de
454 enter price:56
455
456 Enter choice-
457 1) to create
458 2) to create using queue
459 3) to display
460 4) to display using queue
461 5) to insert
462 6) to delete
463 7) to delete using queue
464 8) to sort by price
465 9) to search
466 10) to modify:3
467
468 car_name      Price
469
470 de           56
471
472 we           223
473
474
475 Enter choice-
476 1) to create
477 2) to create using queue
478 3) to display
479 4) to display using queue
480 5) to insert
481 6) to delete
482 7) to delete using queue
483 8) to sort by price
484 9) to search
485 10) to modify:4
486
487 car_name      Price
488
489 er           444
490
491 Enter choice-
492 1) to create
493 2) to create using queue
494 3) to display
495 4) to display using queue
496 5) to insert
497 6) to delete
498 7) to delete using queue
499 8) to sort by price
500 9) to search
501 10) to modify:6
502 Enter car_name you want to delete:de
```

```

503
504 Enter choice-
505     1) to create
506     2) to create using queue
507     3) to display
508     4) to display using queue
509     5) to insert
510     6) to delete
511     7) to delete using queue
512     8) to sort by price
513     9) to search
514    10) to modify:3
515
516 car_name      Price
517
518 we           223
519
520
521 Enter choice-
522     1) to create
523     2) to create using queue
524     3) to display
525     4) to display using queue
526     5) to insert
527     6) to delete
528     7) to delete using queue
529     8) to sort by price
530     9) to search
531    10) to modify:8
532
533 Enter choice-
534     1) to create
535     2) to create using queue
536     3) to display
537     4) to display using queue
538     5) to insert
539     6) to delete
540     7) to delete using queue
541     8) to sort by price
542     9) to search
543    10) to modify:9
544 Enter car_name you want to search:we
545
546 CAR_NAME FOUND
547
548 car_name      Price
549 we           223
550
551 Enter choice-
552     1) to create
553     2) to create using queue
554     3) to display
555     4) to display using queue
556     5) to insert
557     6) to delete
558     7) to delete using queue
559     8) to sort by price
560     9) to search
561    10) to modify:10
562 Enter car_name you want to modify:we
563
564 CAR_NAME FOUND
565
566 car_name      Price
567 we           223
568 enter:
569     1) to modify car_types name
570     2) to modify car_types price:1
571 enter new car_types name:dd
572 press 1 to continue modifying:1
573 enter:
574     1) to modify car_types name
575     2) to modify car_types price:2
576 enter new car_types price:45
577 press 1 to continue modifying:3
578
579 Enter choice-
580     1) to create
581     2) to create using queue
582     3) to display
583     4) to display using queue
584     5) to insert
585     6) to delete
586     7) to delete using queue

```

```
587     8) to sort by price
588     9) to search
589    10) to modify:3
590
591    car_name      Price
592
593    dd           45
594
595
596    Enter choice-
597     1) to create
598     2) to create using queue
599     3) to display
600     4) to display using queue
601     5) to insert
602     6) to delete
603     7) to delete using queue
604     8) to sort by price
605     9) to search
606    10) to modify:
607    */
608
609
610
611
612
613
614
615
616
617
618
619
620
621
```