

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<string.h>
4  #define COUNT 10
5
6  struct car_rent
7  {
8      char car_name[20];
9      char location[20];
10     int price;
11     int km;
12     int time;
13     struct car_rent *link;
14 }*start;
15
16 struct priceTree
17 {
18     int price;
19     struct priceTree *left;
20     struct priceTree *right;
21 };
22 struct car_rent* getNewNode();
23 struct car_rent *create(struct car_rent *start);
24 struct car_rent *insert(struct car_rent *start,int n);
25 struct car_rent *del(struct car_rent *start,char item[]);
26 struct car_rent *sort (struct car_rent *start);
27 struct car_rent *modify(struct car_rent *start,char item[]);
28 void display (struct car_rent *start);
29 void *search(struct car_rent *start,char item[]);
30 void create_queue();
31 void display_queue();
32 void delete_queue();
33 struct car_rent *front=NULL;
34 struct car_rent *rear=NULL;
35 struct priceTree *node;
36 struct priceTree *root;
37 struct priceTree *getTree();
38 struct priceTree *getnode(int x);
39 struct priceTree *BalancedBst(int arr[],int s,int e);
40 void sortTree(int arr[],int n);
41 void print2DUtil(struct priceTree *root, int space);
42 void print2D(struct priceTree *root);
43 struct car_rent *p,*q,*temp;
44 char item[20];
45 int i,n,x,j,ch,l;
46 int s,e,point=0;
47 char password[30]="car";
48 char pass[30];
49 char name[50];
50 int j,pos,choice,ch;
51 int flag=0,chw=0,j=0;
52 void login()
53 {
54     printf("\nEnter your user name:");
55     scanf("%s",name);
56     printf("\nEnter the password:");
57     //scanf("%s",pass);
58     i=0;
59     while(ch!=13)
60     {
61         ch = getch();
62         pass[i]=ch;
63         i++;
64         printf("*");
65     }
66     pass[i-1]='\0';
67     if(stricmp(password,pass)==0)
68     {
69         printf("system is open");
70     }
71     else
72     {
73         printf("\nYour system has been locked");
74         exit (0);
75     }
76 }
77 void main()
78 {
79     login();
80     system("CLS");
81     struct car_rent *start=NULL;
82
83     printf("\t\t*****\n");
84     printf("\t\t*****\n");

```

```

85     printf("\t\t**    CAR RENTAL SYSTEM    **\n");
86     printf("\t\t*****\n");
87     printf("\t\t*****\n");
88     while(1)
89     {
90         printf(" 1) to create\n 2) to display.\n 3) to insert.\n 4) to delete. \n 5) to sort
by price.\n 6) to search. \n 7) to modify. \n 8) create using queue.\n 9) display using
queue.\n 10) delete using queue.\n 11) Create tree of price.\n 12) exit(0).\n\n ");
91         printf("Enter your choice-");
92         scanf("%d",&ch);
93         switch (ch)
94         {
95             case 1:
96                 start=create(start);
97                 break;
98             case 2:
99                 display(start);
100                break;
101            case 3:
102                start=insert(start,n);
103                n++;
104                break;
105            case 4:
106                printf("Enter item you want to delete:");
107                scanf("%s",item);
108                start=del(start,item);
109                break;
110            case 5:
111                sort(start);
112                break;
113            case 6:
114                printf("Enter item you want to search:");
115                scanf("%s",item);
116                search(start,item);
117                break;
118            case 7:
119                printf("Enter item you want to modify:");
120                scanf("%s",item);
121                start=modify(start,item);
122                break;
123            case 8:
124                create_queue();
125                break;
126            case 9:
127                display_queue();
128                break;
129            case 10:
130                delete_queue();
131                break;
132            case 11:
133                print2D(getTree(start));
134                break;
135            case 12:
136                exit(0);
137                break;
138            default:
139                printf("error!");
140        }
141    }
142 }
143 }
144
145 struct car_rent *create(struct car_rent *start)
146 {
147
148     struct car_rent *temp;
149     printf("Enter no of cars you want to rent:");
150     scanf("%d",&n);
151     temp=start;
152     for(i=0;i<n;i++)
153     {
154         temp=(struct car_rent*)malloc(sizeof(struct car_rent)*1);
155         while(1)
156         {
157             printf("\tenter car name:");
158             scanf("%s",(temp->car_name));
159             printf("\tenter location");
160             scanf("%s",(temp->location));
161             l=strlen(temp->car_name);
162             l=strlen(temp->location);
163             if(l > 10 )
164             {
165                 printf("\tPlease enter less characters in list name\n");
166                 printf("\tPlease enter less characters in list name\n");

```

```

167     }
168     else
169     break;
170 }
171
172 do{
173     printf("\ntenter price:");
174     scanf("%d",&(temp->price));
175     if(temp->price > 1 && temp->price < 99999)
176     {
177         break;
178     }
179     else
180     {
181         printf("\tPlease enter less price\n");
182     }
183 }while(1);
184
185 do{
186     printf("\ntenter time:");
187     scanf("%d",&(temp->time));
188     if(temp->time > 1 && temp->time < 55500)
189     {
190         break;
191     }
192     else
193     {
194         printf("\tPlease enter less time\n");
195     }
196 }while(1);
197
198 do{
199
200     printf("\ntenter distance:");
201     scanf("%d",&(temp->km));
202     if(temp-> km > 0 && temp-> km < 22200)
203     {
204         break;
205     }
206     else
207     {
208         printf("\tSORRY..not available Please enter distance between(1 to 1111200)\n");
209     }
210 }while(1);
211 printf("\n");
212
213 if(start==NULL)
214 {
215     start=temp;
216     temp->link=NULL;
217 }
218
219 else
220 {
221     p=start;
222     while(p->link!=NULL)
223     p=p->link;
224     p->link=temp;
225     temp->link=NULL;
226 }
227 }
228 return start;
229 }
230
231 void display (struct car_rent *start)
232 {
233     if(start==NULL)
234     {
235         printf("\n*Order List is Empty*\n\n");
236         return;
237     }
238     p=start;
239     printf("\n\tcar name\tPrice\tlocation\ttime\tdistance.\n");
240     while(p!=NULL)
241     {
242         printf("\n\t%s\t %d\t%s\t%d\t\t%d",p->car_name,p->price,p->location,p->time,p->km);
243         p=p->link;
244     }
245     printf("\n");
246 }
247
248 struct car_rent *insert(struct car_rent *start,int n)
249 {
250     struct car_rent *temp;

```

```

251     int pos, count=0;
252     p=start;
253     temp=(struct car_rent*)malloc(sizeof(struct car_rent)*1);
254     printf("Enter location you want to insert:");
255     scanf("%d", &pos);
256     if(pos<n)
257     {
258         count++;
259     }
260
261     printf("%d\n", n );
262     if(pos==1)
263     {
264         while(1)
265         {
266             printf("\ntenter car name:");
267             scanf("%s", (temp->car_name));
268             printf("\ntenter location");
269             scanf("%s", (temp->location));
270             l=strlen(temp->car_name);
271             l=strlen(temp->location);
272             if(l > 10 )
273             {
274                 printf("\tPlease enter less characters in list name\n");
275                 printf("\tPlease enter less characters in list name\n");
276             }
277             else
278                 break;
279         }
280
281         do{
282             printf("\ntenter price:");
283             scanf("%d", &(temp->price));
284             if(temp->price > 1 && temp->price < 99999)
285             {
286                 break;
287             }
288             else
289             {
290                 printf("\tPlease enter less price\n");
291             }
292         }while(1);
293
294         do{
295             printf("\ntenter time:");
296             scanf("%d", &(temp->time));
297             if(temp->time > 1 && temp->time < 55500)
298             {
299                 break;
300             }
301             else
302             {
303                 printf("\tPlease enter less time\n");
304             }
305         }while(1);
306
307         do{
308             printf("\ntenter distance :");
309             scanf("%d", &(temp->km));
310             if(temp-> km > 0 && temp-> km < 22200)
311             {
312                 break;
313             }
314             else
315             {
316                 printf("\tSORRY..not available Please enter receipt_code between(1 to 200)\n");
317             }
318         }while(1);
319         temp->link=start;
320         start=temp;
321
322         return start;
323     }
324
325     else{
326
327
328
329     for(i=1; i<pos-1 && p!=NULL; i++)
330         p=p->link;
331     if(p==NULL)
332         printf("\n*less no of items*\n\n");
333
334     else

```

```

335 {
336     while(1)
337     {
338         printf("\ntenter car name:");
339         scanf("%s", (temp->car_name));
340         printf("\ntenter location");
341         scanf("%s", (temp->location));
342         l=strlen(temp->car_name);
343         l=strlen(temp->location);
344         if(l > 10 )
345         {
346             printf("\tPlease enter less characters in list name\n");
347             printf("\tPlease enter less characters in list name\n");
348         }
349         else
350             break;
351     }
352
353     do{
354         printf("\ntenter price:");
355         scanf("%d",&(temp->price));
356         if(temp->price > 1 && temp->price < 99999)
357         {
358             break;
359         }
360         else
361         {
362             printf("\tPlease enter less price\n");
363         }
364     }while(1);
365
366     do{
367         printf("\ntenter time:");
368         scanf("%d",&(temp->time));
369         if(temp->time > 1 && temp->time < 55500)
370         {
371             break;
372         }
373         else
374         {
375             printf("\tPlease enter less time\n");
376         }
377     }while(1);
378     do{
379         printf("\ntenter distance :");
380         scanf("%d",&(temp->km));
381         if(temp-> km > 0 && temp-> km < 22200)
382         {
383             break;
384         }
385         else
386         {
387             printf("\tSORRY..not available Please enter receipt_code between(1 to 200)\n");
388         }
389     }while(1);
390     temp->link=p->link;
391     p->link=temp;
392 }
393 return start;
394 }
395
396
397 struct car_rent *del(struct car_rent *start,char item[])
398 {
399     struct car_rent *temp;
400     if(strcmp(start->car_name,item)==0)
401     {
402         temp=start;
403         start=temp->link;
404         free(temp);
405         return start;
406     }
407     p=start;
408
409     while(p->link!=NULL)
410     {
411         if(strcmp(p->link->car_name,item)==0)
412         {
413             temp=p->link;
414             p->link=temp->link;
415             free(temp);
416             return start;
417         }
418         p=p->link;

```



```

500         scanf("%s", (temp->car_name));
501         l=strlen(temp->car_name);
502         if(l > 10 )
503         {
504             printf("\tPlease enter less characters in list name\n");
505         }
506         else
507         {
508             break;
509         }
510     }
511     break;
512
513 case 2:
514     do{
515         printf("\tenter price:");
516         scanf("%d",&(temp->price));
517         if(temp->price > 1 && temp->price < 99999)
518         {
519             break;
520         }
521         else
522         {
523             printf("\tPlease enter less price\n");
524         }
525     }while(1);
526     break;
527
528 }
529
530 printf("press 1 to continue modifying:");
531 scanf("%d",&chw);
532 }while(chw==1);
533
534     flag++;
535     return start;
536 }
537
538 p=start;
539
540 while(p->link!=NULL)
541 {
542     if(strcmp(p->link->car_name,item)==0)
543     {
544         printf("\n*ITEM FOUND\n");
545         printf("\ncar name\tPrice\tlocation\ttime\tdistance");
546         printf("\n%s\t %d\t %s\t %d\t %d\n",p->link->car_name,p->link->price,p->link->location,p->link->time,p->link->km);
547
548         do
549         {
550             printf("enter:\n 1) to modify car name\n 2) to modify list price:");
551             scanf("%d",&ch);
552
553             switch (ch)
554             {
555                 case 1:
556                     while(1)
557                     {
558                         printf("\tenter car name:");
559                         scanf("%s", (temp->car_name));
560                         l=strlen(temp->car_name);
561                         if(l > 10 )
562                         {
563                             printf("\tPlease enter less characters in list name\n");
564                         }
565                         else
566                         {
567                             break;
568                         }
569                     }
570                     break;
571
572                 case 2:
573                     do{
574                         printf("\tenter price:");
575                         scanf("%d",&(temp->price));
576                         if(temp->price > 1 && temp->price < 99999)
577                         {
578                             break;
579                         }
580                         else
581                         {
582                             printf("\tPlease enter less price\n");

```

```

583         }
584         }while(1);
585         break;
586     }
587     printf("press 1 to continue modifying:");
588     scanf("%d",&chw);
589     }while(chw==1);
590
591
592
593     }
594     p=p->link;
595     flag++;
596     return start;
597 }
598
599 if(flag==0)
600 printf("\n\nnot found\n\n");
601 }
602
603 struct car_rent* getNewNode()
604 {
605     return (struct car_rent*)malloc(sizeof(struct car_rent));
606 }
607 void create_queue()
608 {
609     int q;
610     struct car_rent *temp;
611     printf("Enter no.of type of item:");
612     scanf("%d",&q);
613     for(i=0;i<n;i++)
614     {
615         temp=(struct car_rent*)malloc(sizeof(struct car_rent)*1);
616         while(1)
617         {
618             printf("\ntenter car name:");
619             scanf("%s", (temp->car_name));
620             printf("\ntenter location");
621             scanf("%s", (temp->location));
622             l=strlen(temp->car_name);
623             l=strlen(temp->location);
624             if(l > 10 )
625             {
626                 printf("\tPlease enter less characters in list name\n");
627                 printf("\tPlease enter less characters in list name\n");
628             }
629             else
630                 break;
631         }
632
633         do{
634             printf("\ntenter price:");
635             scanf("%d",&(temp->price));
636             if(temp->price > 1 && temp->price < 99999)
637             {
638                 break;
639             }
640             else
641             {
642                 printf("\tPlease enter less price\n");
643             }
644         }while(1);
645
646         do{
647             printf("\ntenter time:");
648             scanf("%d",&(temp->time));
649             if(temp->time > 1 && temp->time < 55500)
650             {
651                 break;
652             }
653             else
654             {
655                 printf("\tPlease enter less time\n");
656             }
657         }while(1);
658
659         do{
660
661             printf("\ntenter distance:");
662             scanf("%d",&(temp->km));
663             if(temp-> km > 0 && temp-> km < 22200)
664             {
665                 break;
666             }

```



```

667         else
668         {
669             printf("\tSORRY..not available Please enter distance between(1 to 1111200)\n");
670         }
671     }while(1);
672     printf("\n");
673
674     if (rear==NULL)
675     {
676         rear= temp;
677         front = temp;
678         rear->link=NULL;
679         continue;
680     }
681     rear->link=temp;
682     rear=temp;
683 }
684 }
685 void display_queue()
686 {
687     if(front==NULL)
688     {
689         printf("\n No ITEM IN LIST\n");
690     }
691     else
692     {
693         p=front;
694         printf("\n\tcar name \tPRICE \tlocation \ttime \tdistance \n\n");
695         while(p!=NULL)
696         {
697             printf("\t%s\t%d\t%s\t%d\t%d",p->car_name,p->price,p->location,p->time,p->km);
698             printf("\n");
699             p=p->link;
700         }
701     }
702 }
703 void delete_queue()
704 {
705     if(front==NULL)
706     {
707         printf("\n*Queue Underflow*\n");
708     }
709     else
710     {
711         temp=front;
712         front=front->link;
713         free(temp);
714     }
715 }
716 struct priceTree *getTree(struct car_rent *start)
717 {
718     int arr[n];
719     struct car_rent *temp=start;
720
721     while(temp!=NULL)
722     {
723         arr[point++]=temp->price;
724         temp=temp->link;
725     }
726     sortTree(arr,n);
727     return BalancedBst(arr,0,n-1);
728 }
729 void sortTree(int arr[],int n)
730 {
731     int temp;
732     for(int i = 0; i < n; i++)
733     {
734         for(int j = 0; j < (n-1-i); j++)
735         {
736             if(arr[j] > arr[j+1]) //for decending order change "arr[j]<arr[j+1]"
737             {
738                 temp = arr[j];
739                 arr[j] = arr[j+1];
740                 arr[j+1] = temp;
741             }
742         }
743     }
744 }
745
746 struct priceTree *getnode(int x)
747 {
748     struct priceTree *temp=(struct priceTree*)malloc(sizeof(struct priceTree));
749     temp->price=x;
750     temp->left=temp->right=NULL;

```

```

751     return temp;
752 }
753 struct priceTree *BalancedBst(int arr[],int s,int e)
754 {
755     if(s > e)
756         return NULL;
757
758     int mid=(s+e)/2;
759     struct priceTree *root=getnode(arr[mid]);
760     root->left=BalancedBst(arr,s,mid-1);
761     root->right=BalancedBst(arr,mid+1,e);
762     return root;
763 }
764
765 void print2DUtil(struct priceTree *root, int space)
766 {
767
768     if (root == NULL)
769         return;
770     space+=COUNT;
771     print2DUtil(root->right, space);
772
773     printf("\n");
774     for (int i = COUNT; i < space; i++)
775         printf(" ");
776     printf("%d\n\n", root->price);
777
778     print2DUtil(root->left, space);
779 }
780
781 void print2D(struct priceTree *root)
782 {
783     print2DUtil(root, 0);
784 }
785
786
787
788 /*OUTPUT
789 Enter your user name:car
790
791 Enter the password:***
792
793 *****
794 *****
795 **   CAR RENTAL SYSTEM   **
796 *****
797 *****
798 1) to create
799 2) to display.
800 3) to insert.
801 4) to delete.
802 5) to sort by price.
803 6) to search.
804 7) to modify.
805 8) create using queue.
806 9) display using queue.
807 10) delete using queue.
808 11) Create tree of price.
809 12) exit(0).
810
811 Enter your choice-*/
812

```