

```

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 int flag=0;
48 void main()
49 {
50     struct car_rent *start=NULL;
51
52     printf("\t\t*****\n");
53     printf("\t\t*****\n");
54     printf("\t\t**   CAR RENTAL SYSTEM   **\n");
55     printf("\t\t*****\n");
56     printf("\t\t*****\n");
57     while(1)
58     {
59         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 ");
60         printf("Enter your choice-");
61         scanf("%d",&ch);
62         switch (ch)
63         {
64             case 1:
65                 start=create(start);
66                 break;
67             case 2:
68                 display(start);
69                 break;
70             case 3:
71                 start=insert(start,n);
72                 n++;
73                 break;
74             case 4:
75                 printf("Enter item you want to delete:");
76                 scanf("%s",item);
77                 start=del(start,item);
78                 break;
79             case 5:
80                 sort(start);
81                 break;
82             case 6:

```

```

83         printf("Enter item you want to search:");
84         scanf("%s", item);
85         search(start, item);
86         break;
87     case 7:
88         printf("Enter item you want to modify:");
89         scanf("%s", item);
90         start=modify(start, item);
91         break;
92     case 8:
93         create_queue();
94         break;
95     case 9:
96         display_queue();
97         break;
98     case 10:
99         delete_queue();
100        break;
101    case 11:
102        print2D(getTree(start));
103        break;
104    case 12:
105        exit(0);
106        break;
107    default:
108        printf("error!");
109    }
110 }
111 }
112 }
113
114 struct car_rent *create(struct car_rent *start)
115 {
116
117     struct car_rent *temp;
118     printf("Enter no of cars you want to rent:");
119     scanf("%d", &n);
120     temp=start;
121     for(i=0; i<n; i++)
122     {
123         temp=(struct car_rent*)malloc(sizeof(struct car_rent)*1);
124         while(1)
125         {
126             printf("\tenter car name:");
127             scanf("%s", (temp->car_name));
128             printf("\tenter location");
129             scanf("%s", (temp->location));
130             l=strlen(temp->car_name);
131             l=strlen(temp->location);
132             if(l > 10 )
133             {
134                 printf("\tPlease enter less characters in list name\n");
135                 printf("\tPlease enter less characters in list name\n");
136             }
137             else
138                 break;
139         }
140
141         do{
142             printf("\tenter price:");
143             scanf("%d", &(temp->price));
144             if(temp->price > 1 && temp->price < 99999)
145             {
146                 break;
147             }
148             else
149             {
150                 printf("\tPlease enter less price\n");
151             }
152         }while(1);
153
154         do{
155             printf("\tenter time:");
156             scanf("%d", &(temp->time));
157             if(temp->time > 1 && temp->time < 55500)
158             {
159                 break;
160             }
161             else
162             {
163                 printf("\tPlease enter less time\n");
164             }
165         }while(1);
166

```

```

167     do{
168         printf("\ntenter distance:");
169         scanf("%d",&(temp->km));
170         if(temp-> km > 0 && temp-> km < 22200)
171         {
172             break;
173         }
174         else
175         {
176             printf("\tSORRY..not available Please enter distance between(1 to 11111200)\n");
177         }
178     }while(1);
179     printf("\n");
180
181     if(start==NULL)
182     {
183         start=temp;
184         temp->link=NULL;
185     }
186
187     else
188     {
189         p=start;
190         while(p->link!=NULL)
191         p=p->link;
192         p->link=temp;
193         temp->link=NULL;
194     }
195 }
196
197 return start;
198 }
199
200 void display (struct car_rent *start)
201 {
202     if(start==NULL)
203     {
204         printf("\n*Order List is Empty*\n\n");
205         return;
206     }
207     p=start;
208     printf("\n\tcar name\tPrice\tlocation\ttime\tdistance.\n");
209     while(p!=NULL)
210     {
211         printf("\n\t%s\t %d\t%s\t%d\t\t%d",p->car_name,p->price,p->location,p->time,p->km);
212         p=p->link;
213     }
214     printf("\n");
215 }
216
217 struct car_rent *insert(struct car_rent *start,int n)
218 {
219     struct car_rent *temp;
220     int pos,count=0;
221     p=start;
222     temp=(struct car_rent*)malloc(sizeof(struct car_rent)*1);
223     printf("Enter location you want to insert:");
224     scanf("%d",&pos);
225     if(pos<n)
226     {
227         count++;
228     }
229
230     printf("%d\n",n );
231     if(pos==1)
232     {
233         while(1)
234         {
235             printf("\ntenter car name:");
236             scanf("%s",(temp->car_name));
237             printf("\ntenter location");
238             scanf("%s",(temp->location));
239             l=strlen(temp->car_name);
240             l=strlen(temp->location);
241             if(l > 10 )
242             {
243                 printf("\tPlease enter less characters in list name\n");
244                 printf("\tPlease enter less characters in list name\n");
245             }
246             else
247                 break;
248         }
249     }
250     do{

```

```

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

```

```

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

```

```

419     }
420 }
421
422 void *search(struct car_rent *start, char item[])
423 {
424     if(strcmp(start->car_name, item)==0)
425     {
426         printf("\nITEM FOUND\n");
427         printf("\ncar name\tPrice\tlocation\ttime\tdistance\n.");
428         printf("\n%s\t %d\t %s\t %d\t %d\n", start->car_name, start->price, start->location, start->time, start->km);
429         flag++;
430     }
431     p=start;
432     while(p->link!=NULL)
433     {
434         if(strcmp(p->link->car_name, item)==0)
435         {
436             printf("\n*ITEM FOUND\n");
437             printf("\ncar name\tPrice\tlocation\ttime\tdistance\n.");
438             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);
439         }
440         p=p->link;
441         flag++;
442     }
443
444     if(flag==0)
445         printf("\n\nnot found\n\n");
446 }
447
448
449 struct car_rent *modify(struct car_rent *start, char item[])
450 {
451     char ans1, ans2;
452     int chw=0;
453     if(strcmp(start->car_name, item)==0)
454     {
455         printf("\nITEM FOUND\n");
456         printf("\ncar name\tPrice\tlocation\ttime\tdistance\n.");
457         printf("\n%s\t %d\t %s\t %d\t %d\n", start->car_name, start->price, start->location, start->time, start->km);
458
459         do
460         {
461             printf("enter:\n 1) to modify car name\n 2) to modify list price:");
462             scanf("%d", &ch);
463             switch (ch)
464             {
465                 case 1:
466                     while(1)
467                     {
468                         printf("\tentent product name:");
469                         scanf("%s", (temp->car_name));
470                         l=strlen(temp->car_name);
471                         if(l > 10 )
472                         {
473                             printf("\tPlease enter less characters in list name\n");
474                         }
475                         else
476                         {
477                             break;
478                         }
479                     }
480                     break;
481
482                 case 2:
483                     do{
484                         printf("\tentent price:");
485                         scanf("%d", &(temp->price));
486                         if(temp->price > 1 && temp->price < 99999)
487                         {
488                             break;
489                         }
490                         else
491                         {
492                             printf("\tPlease enter less price\n");
493                         }
494                     }while(1);
495                     break;
496             }
497         }
498
499         printf("press 1 to continue modifying:");

```

```

500     scanf("%d",&chw);
501 }while(chw==1);
502
503     flag++;
504     return start;
505 }
506
507 p=start;
508
509 while(p->link!=NULL)
510 {
511     if(strcmp(p->link->car_name,item)==0)
512     {
513         printf("\n*ITEM FOUND\n");
514         printf("\ncar name\tPrice\tlocation\ttime\tdistance");
515         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);
516
517         do
518         {
519             printf("enter:\n 1) to modify car name\n 2) to modify list price:");
520             scanf("%d",&ch);
521
522             switch (ch)
523             {
524                 case 1:
525                     while(1)
526                     {
527                         printf("\ntenter car name:");
528                         scanf("%s", (temp->car_name));
529                         l=strlen(temp->car_name);
530                         if(l > 10 )
531                         {
532                             printf("\tPlease enter less characters in list name\n");
533                         }
534                         else
535                         {
536                             break;
537                         }
538                     }
539                     break;
540
541                 case 2:
542                     do{
543                         printf("\ntenter price:");
544                         scanf("%d",&(temp->price));
545                         if(temp->price > 1 && temp->price < 99999)
546                         {
547                             break;
548                         }
549                         else
550                         {
551                             printf("\tPlease enter less price\n");
552                         }
553                     }while(1);
554                     break;
555
556             }
557             printf("press 1 to continue modifying:");
558             scanf("%d",&chw);
559             }while(chw==1);
560
561         }
562
563         p=p->link;
564         flag++;
565         return start;
566     }
567
568     if(flag==0)
569         printf("\n\nnot found\n\n");
570 }
571
572 struct car_rent* getNewNode()
573 {
574     return (struct car_rent*)malloc(sizeof(struct car_rent));
575 }
576 void create_queue()
577 {
578     int q;
579     struct car_rent *temp;
580     printf("Enter no.of type of item:");
581     scanf("%d",&q);
582     for(i=0;i<n;i++)

```

```

583 {
584     temp=(struct car_rent*)malloc(sizeof(struct car_rent)*1);
585     while(1)
586     {
587         printf("\ntenter car name:");
588         scanf("%s", (temp->car_name));
589         printf("\ntenter location");
590         scanf("%s", (temp->location));
591         l=strlen(temp->car_name);
592         l=strlen(temp->location);
593         if(l > 10 )
594         {
595             printf("\tPlease enter less characters in list name\n");
596             printf("\tPlease enter less characters in list name\n");
597         }
598         else
599             break;
600     }
601
602     do{
603         printf("\ntenter price:");
604         scanf("%d",&(temp->price));
605         if(temp->price > 1 && temp->price < 99999)
606         {
607             break;
608         }
609         else
610         {
611             printf("\tPlease enter less price\n");
612         }
613     }while(1);
614
615     do{
616         printf("\ntenter time:");
617         scanf("%d",&(temp->time));
618         if(temp->time > 1 && temp->time < 55500)
619         {
620             break;
621         }
622         else
623         {
624             printf("\tPlease enter less time\n");
625         }
626     }while(1);
627
628     do{
629
630         printf("\ntenter distance:");
631         scanf("%d",&(temp->km));
632         if(temp-> km > 0 && temp-> km < 22200)
633         {
634             break;
635         }
636         else
637         {
638             printf("\tSORRY..not available Please enter distance between(1 to 11111200)\n");
639         }
640     }while(1);
641     printf("\n");
642
643     if (rear==NULL)
644     {
645         rear= temp;
646         front = temp;
647         rear->link=NULL;
648         continue;
649     }
650     rear->link=temp;
651     rear=temp;
652 }
653 }
654 void display_queue()
655 {
656     if(front==NULL)
657     {
658         printf("\n No ITEM IN LIST\n");
659     }
660     else
661     {
662         p=front;
663         printf("\n\tcar name \tPRICE \tlocation \ttime \tdistance \n\n");
664         while(p!=NULL)
665         {
666             printf("\t%s\t%d\t%s\t%d\t%d",p->car_name,p->price,p->location,p->time,p->km);

```



```

667         printf("\n");
668         p=p->link;
669     }
670 }
671 }
672 void delete_queue()
673 {
674     if(front==NULL)
675     {
676         printf("\n*Queue Underflow*\n");
677     }
678     else
679     {
680         temp=front;
681         front=front->link;
682         free(temp);
683     }
684 }
685 struct priceTree *getTree(struct car_rent *start)
686 {
687     int arr[n];
688     struct car_rent *temp=start;
689
690     while(temp!=NULL)
691     {
692         arr[point++]=temp->price;
693         temp=temp->link;
694     }
695     sortTree(arr,n);
696     return BalancedBst(arr,0,n-1);
697 }
698 void sortTree(int arr[],int n)
699 {
700     int temp;
701     for(int i = 0; i < n; i++)
702     {
703         for(int j = 0; j < (n-1-i); j++)
704         {
705             if(arr[j] > arr[j+1]) //for decending order change "arr[j]<arr[j+1]"
706             {
707                 temp = arr[j];
708                 arr[j] = arr[j+1];
709                 arr[j+1] = temp;
710             }
711         }
712     }
713 }
714 }
715 struct priceTree *getnode(int x)
716 {
717     struct priceTree *temp=(struct priceTree*)malloc(sizeof(struct priceTree));
718     temp->price=x;
719     temp->left=temp->right=NULL;
720     return temp;
721 }
722 struct priceTree *BalancedBst(int arr[],int s,int e)
723 {
724     if(s > e)
725         return NULL;
726
727     int mid=(s+e)/2;
728     struct priceTree *root=getnode(arr[mid]);
729     root->left=BalancedBst(arr,s,mid-1);
730     root->right=BalancedBst(arr,mid+1,e);
731     return root;
732 }
733
734 void print2DUtil(struct priceTree *root, int space)
735 {
736
737     if (root == NULL)
738         return;
739     space+=COUNT;
740     print2DUtil(root->right, space);
741
742     printf("\n");
743     for (int i = COUNT; i < space; i++)
744         printf(" ");
745     printf("%d\n\n", root->price);
746
747     print2DUtil(root->left, space);
748 }
749
750 void print2D(struct priceTree *root)

```

```

751 {
752     print2DUtil(root, 0);
753 }
754
755
756
757
758
759
760 /* OUTPUT
761 *****
762
763             *****
764             **      CAR RENTAL SYSTEM      **
765             *****
766
767 1) to create
768 2) to display.
769 3) to insert.
770 4) to delete.
771 5) to sort by price.
772 6) to search.
773 7) to modify.
774 8) create using queue.
775 9) display using queue.
776 10) delete using queue.
777 11) Create tree of price.
778 12) exit(0).
779
780 Enter your choice-1
781 Enter no of cars you want to rent:1
782     enter car name:LAMBORGANILAND
783     enter locationMUMBAIMAHARASTRA
784     Please enter less characters in list name
785     Please enter less characters in list name
786     enter car name:LAMBORGANI
787     enter locationMUMBAI
788     enter price:650000000000
789     Please enter less price
790     enter price:6500
791     enter time:750000000000
792     Please enter less time
793     enter time:750
794     enter distance:893365666
795     SORRY..not available Please enter distance between(1 to 11111200)
796     enter distance:890
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-2
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834 */

```

car name	Price	location	time	distance.
LAMBORGANI	6500	MUMBAI	750	890

835
836
837
838
839