

```

1  ASSIGNMENT 8
2  #include<stdio.h>
3  #include<stdlib.h>
4  #include<string.h>
5  #define COUNT 10
6
7  struct car_rent
8  {
9      char car_name[20];
10     char location[20];
11     int price;
12     int km;
13     int time;
14     struct car_rent *link;
15 }*start;
16
17 struct priceTree
18 {
19     int price;
20     struct priceTree *left;
21     struct priceTree *right;
22 };
23 struct car_rent* getNewNode();
24 struct car_rent *create(struct car_rent *start);
25 struct car_rent *insert(struct car_rent *start,int n);
26 struct car_rent *del(struct car_rent *start,char item[]);
27 struct car_rent *sort (struct car_rent *start);
28 struct car_rent *modify(struct car_rent *start,char item[]);
29 void display (struct car_rent *start);
30 void *search(struct car_rent *start,char item[]);
31 void create_queue();
32 void display_queue();
33 void delete_queue();
34 struct car_rent *front=NULL;
35 struct car_rent *rear=NULL;
36 struct priceTree *node;
37 struct priceTree *root;
38 struct priceTree *getTree();
39 struct priceTree *getnode(int x);
40 struct priceTree *BalancedBst(int arr[],int s,int e);
41 void sortTree(int arr[],int n);
42 void print2DUtil(struct priceTree *root, int space);
43 void print2D(struct priceTree *root);
44 struct car_rent *p,*q,*temp;
45 char item[20];
46 int i,n,x,j,ch,l;
47 int s,e,point=0;
48 int flag=0;
49 void main()
50 {
51     struct car_rent *start=NULL;
52
53     printf("\t\t*****\n");
54     printf("\t\t*****\n");
55     printf("\t\t**    CAR RENTAL SYSTEM    **\n");
56     printf("\t\t*****\n");
57     printf("\t\t*****\n");
58     while(1)
59     {
60         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 ");
61         printf("Enter your choice-");
62         scanf("%d",&ch);
63         switch (ch)
64         {
65             case 1:
66                 start=create(start);
67                 break;
68             case 2:
69                 display(start);
70                 break;
71             case 3:
72                 start=insert(start,n);
73                 n++;
74                 break;
75             case 4:
76                 printf("Enter item you want to delete:");
77                 scanf("%s",item);
78                 start=del(start,item);
79                 break;
80             case 5:
81                 sort(start);
82                 break;

```

```

83     case 6:
84         printf("Enter item you want to search:");
85         scanf("%s", item);
86         search(start, item);
87         break;
88     case 7:
89         printf("Enter item you want to modify:");
90         scanf("%s", item);
91         start=modify(start, item);
92         break;
93     case 8:
94         create_queue();
95         break;
96     case 9:
97         display_queue();
98         break;
99     case 10:
100        delete_queue();
101        break;
102    case 11:
103        print2D(getTree(start));
104        break;
105    case 12:
106        exit(0);
107        break;
108    default:
109        printf("error!");
110    }
111 }
112 }
113 }
114
115 struct car_rent *create(struct car_rent *start)
116 {
117
118     struct car_rent *temp;
119     printf("Enter no of cars you want to rent:");
120     scanf("%d", &n);
121     temp=start;
122     for(i=0; i<n; i++)
123     {
124         temp=(struct car_rent*)malloc(sizeof(struct car_rent)*1);
125         while(1)
126         {
127             printf("\tenter car name:");
128             scanf("%s", (temp->car_name));
129             printf("\tenter location");
130             scanf("%s", (temp->location));
131             l=strlen(temp->car_name);
132             l=strlen(temp->location);
133             if(l > 10 )
134             {
135                 printf("\tPlease enter less characters in list name\n");
136                 printf("\tPlease enter less characters in list name\n");
137             }
138             else
139                 break;
140         }
141
142         do{
143             printf("\tenter price:");
144             scanf("%d", &(temp->price));
145             if(temp->price > 1 && temp->price < 99999)
146             {
147                 break;
148             }
149             else
150             {
151                 printf("\tPlease enter less price\n");
152             }
153         }while(1);
154
155         do{
156             printf("\tenter time:");
157             scanf("%d", &(temp->time));
158             if(temp->time > 1 && temp->time < 55500)
159             {
160                 break;
161             }
162             else
163             {
164                 printf("\tPlease enter less time\n");
165             }
166         }while(1);

```

```

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
198 return start;
199 }
200
201 void display (struct car_rent *start)
202 {
203     if(start==NULL)
204     {
205         printf("\n*Order List is Empty*\n\n");
206         return;
207     }
208     p=start;
209     printf("\n\tcar name\tPrice\tlocation\ttime\tdistance.\n");
210     while(p!=NULL)
211     {
212         printf("\n\t%s\t %d\t%s\t%d\t\t%d",p->car_name,p->price,p->location,p->time,p->km);
213         p=p->link;
214     }
215     printf("\n");
216 }
217
218 struct car_rent *insert(struct car_rent *start,int n)
219 {
220     struct car_rent *temp;
221     int pos,count=0;
222     p=start;
223     temp=(struct car_rent*)malloc(sizeof(struct car_rent)*1);
224     printf("Enter location you want to insert:");
225     scanf("%d",&pos);
226     if(pos<n)
227     {
228         count++;
229     }
230
231     printf("%d\n",n );
232     if(pos==1)
233     {
234         while(1)
235         {
236             printf("\ntenter car name:");
237             scanf("%s", (temp->car_name));
238             printf("\ntenter location");
239             scanf("%s", (temp->location));
240             l=strlen(temp->car_name);
241             l=strlen(temp->location);
242             if(l > 10 )
243             {
244                 printf("\tPlease enter less characters in list name\n");
245                 printf("\tPlease enter less characters in list name\n");
246             }
247             else
248                 break;
249         }
250     }

```

```

251     do{
252         printf("\ntenter price:");
253         scanf("%d",&(temp->price));
254         if(temp->price > 1 && temp->price < 99999)
255         {
256             break;
257         }
258         else
259         {
260             printf("\tPlease enter less price\n");
261         }
262     }while(1);
263
264     do{
265         printf("\ntenter time:");
266         scanf("%d",&(temp->time));
267         if(temp->time > 1 && temp->time < 55500)
268         {
269             break;
270         }
271         else
272         {
273             printf("\tPlease enter less time\n");
274         }
275     }while(1);
276     do{
277         printf("\ntenter 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
295 else{
296
297
298
299     for(i=1;i<pos-1 && p!=NULL;i++)
300     p=p->link;
301     if(p==NULL)
302     printf("\n*less no of items*\n\n");
303
304     else
305     {
306         while(1)
307         {
308             printf("\ntenter car name:");
309             scanf("%s", (temp->car_name));
310             printf("\ntenter location");
311             scanf("%s", (temp->location));
312             l=strlen(temp->car_name);
313             l=strlen(temp->location);
314             if(l > 10 )
315             {
316                 printf("\tPlease enter less characters in list name\n");
317                 printf("\tPlease enter less characters in list name\n");
318             }
319             else
320             break;
321         }
322
323         do{
324             printf("\ntenter price:");
325             scanf("%d",&(temp->price));
326             if(temp->price > 1 && temp->price < 99999)
327             {
328                 break;
329             }
330             else
331             {
332                 printf("\tPlease enter less price\n");
333             }
334         }while(1);

```

```

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

```

```

419     }
420   }
421 }
422 
423 void *search(struct car_rent *start,char item[])
424 {
425     if(strcmp(start->car_name,item)==0)
426     {
427         printf("\nITEM FOUND\n");
428         printf("ncar name\tPrice\tlocation\time\tdistance\n.");
429         printf("%d\n",start->car_name,start->price,start->location,start->time,start->km);
430         flag++;
431     }
432     p=start;
433     while(p->link!=NULL)
434     {
435         if(strcmp(p->link->car_name,item)==0)
436         {
437             printf("\n*ITEM FOUND\n");
438             printf("ncar name\tPrice\tlocation\time\tdistance");
439             printf("%d\n",p->link->car_name,p->link->price,p->link->location,p->link->time,p->link->km);
440         }
441         p=p->link;
442         flag++;
443     }
444 
445     if(flag==0)
446         printf("\nnnot found\n\n");
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\time\tdistance\n.");
457         printf("%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("\ntenter 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                 case 2:
482                     do{
483                         printf("\ntenter price:");
484                         scanf("%d",&(temp->price));
485                         if(temp->price > 1 && temp->price < 99999)
486                         {
487                             break;
488                         }
489                         else
490                         {
491                             printf("\tPlease enter less price\n");
492                         }
493                     }while(1);
494                     break;
495             }

```

```

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

```

```

583     for(i=0;i<n;i++)
584     {
585         temp=(struct car_rent*)malloc(sizeof(struct car_rent)*1);
586         while(1)
587         {
588             printf("\ntenter car name:");
589             scanf("%s", (temp->car_name));
590             printf("\ntenter location");
591             scanf("%s", (temp->location));
592             l=strlen(temp->car_name);
593             l=strlen(temp->location);
594             if(l > 10 )
595             {
596                 printf("\tPlease enter less characters in list name\n");
597                 printf("\tPlease enter less characters in list name\n");
598             }
599             else
600             break;
601         }
602
603         do{
604             printf("\ntenter price:");
605             scanf("%d",&(temp->price));
606             if(temp->price > 1 && temp->price < 99999)
607             {
608                 break;
609             }
610             else
611             {
612                 printf("\tPlease enter less price\n");
613             }
614         }while(1);
615
616         do{
617             printf("\ntenter time:");
618             scanf("%d",&(temp->time));
619             if(temp->time > 1 && temp->time < 55500)
620             {
621                 break;
622             }
623             else
624             {
625                 printf("\tPlease enter less time\n");
626             }
627         }while(1);
628
629         do{
630
631             printf("\ntenter distance:");
632             scanf("%d",&(temp->km));
633             if(temp-> km > 0 && temp-> km < 22200)
634             {
635                 break;
636             }
637             else
638             {
639                 printf("\tSORRY..not available Please enter distance between(1 to 1111200)\n");
640             }
641         }while(1);
642         printf("\n");
643
644         if (rear==NULL)
645         {
646             rear= temp;
647             front = temp;
648             rear->link=NULL;
649             continue;
650         }
651         rear->link=temp;
652         rear=temp;
653     }
654 }
655 void display_queue()
656 {
657     if(front==NULL)
658     {
659         printf("\n No ITEM IN LIST\n");
660     }
661     else
662     {
663         p=front;
664         printf("\n\tcar name \tPRICE \tlocation \ttime \tdistance \n\n");
665         while(p!=NULL)
666         {

```



```

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

```

```

751 void print2D(struct priceTree *root)
752 {
753     print2DUtil(root, 0);
754 }
755
756
757
758
759
760
761
762
763
764
765
766
767
768 /*
769
770 OUTPUT
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834

```

```

*****
*****
**   CAR RENTAL SYSTEM   **
*****
*****
1) to create
2) to display.
3) to insert.
4) to delete.
5) to sort by price.
6) to search.
7) to modify.
8) create using queue.
9) display using queue.
10) delete using queue.
11) Create tree of price.
12) exit(0).

Enter your choice-2

*Order List is Empty*

1) to create
2) to display.
3) to insert.
4) to delete.
5) to sort by price.
6) to search.
7) to modify.
8) create using queue.
9) display using queue.
10) delete using queue.
11) Create tree of price.
12) exit(0).

Enter your choice-1
Enter no of cars you want to rent:2
    enter car name:audi
    enter locationpune
    enter price:23
    enter time:34
    enter distance:64

    enter car name:bmw
    enter locationmumbai
    enter price:4225
    enter time:43
    enter distance:53

1) to create
2) to display.
3) to insert.
4) to delete.
5) to sort by price.
6) to search.
7) to modify.
8) create using queue.
9) display using queue.
10) delete using queue.
11) Create tree of price.
12) exit(0).

Enter your choice-2

```

```

835         car name      Price  location      time  distance.
836
837         audi      23      pune      34      64
838         bmw      4225    mumbai    43      53
839     1) to create
840     2) to display.
841     3) to insert.
842     4) to delete.
843     5) to sort by price.
844     6) to search.
845     7) to modify.
846     8) create using queue.
847     9) display using queue.
848     10) delete using queue.
849     11) Create tree of price.
850     12) exit(0).
851
852     Enter your choice-3
853     Enter location you want to insert:1
854     2
855         enter car name:maruti
856         enter locationsonari
857         enter price:2424
858         enter time:24
859         enter distance :522
860     1) to create
861     2) to display.
862     3) to insert.
863     4) to delete.
864     5) to sort by price.
865     6) to search.
866     7) to modify.
867     8) create using queue.
868     9) display using queue.
869     10) delete using queue.
870     11) Create tree of price.
871     12) exit(0).
872
873     Enter your choice-2
874
875         car name      Price  location      time  distance.
876
877         maruti    2424    sonari    24      522
878         audi      23      pune      34      64
879         bmw      4225    mumbai    43      53
880     1) to create
881     2) to display.
882     3) to insert.
883     4) to delete.
884     5) to sort by price.
885     6) to search.
886     7) to modify.
887     8) create using queue.
888     9) display using queue.
889     10) delete using queue.
890     11) Create tree of price.
891     12) exit(0).
892
893     Enter your choice-4
894     Enter item you want to delete:bmw
895     1) to create
896     2) to display.
897     3) to insert.
898     4) to delete.
899     5) to sort by price.
900     6) to search.
901     7) to modify.
902     8) create using queue.
903     9) display using queue.
904     10) delete using queue.
905     11) Create tree of price.
906     12) exit(0).
907
908     Enter your choice-2
909
910         car name      Price  location      time  distance.
911
912         maruti    2424    sonari    24      522
913         audi      23      pune      34      64
914
915     Enter your choice-5
916     1) to create
917     2) to display.
918     3) to insert.

```

```

919 4) to delete.
920 5) to sort by price.
921 6) to search.
922 7) to modify.
923 8) create using queue.
924 9) display using queue.
925 10) delete using queue.
926 11) Create tree of price.
927 12) exit(0).
928
929 Enter your choice-2
930
931      car name      Price  location      time  distance.
932
933      audi      23      pune      34      64
934      maruti    2424    sonari    24      522
935 1) to create
936 2) to display.
937 3) to insert.
938 4) to delete.
939 5) to sort by price.
940 6) to search.
941 7) to modify.
942 8) create using queue.
943 9) display using queue.
944 10) delete using queue.
945 11) Create tree of price.
946 12) exit(0).
947
948 Enter your choice-6
949 Enter item you want to search:audi
950
951 ITEM FOUND
952
953 car name      Price  location      time  distance
954 .
955 audi      23      pune      34      64
956 1) to create
957 2) to display.
958 3) to insert.
959 4) to delete.
960 5) to sort by price.
961 6) to search.
962 7) to modify.
963 8) create using queue.
964 9) display using queue.
965 10) delete using queue.
966 11) Create tree of price.
967 12) exit(0).
968
969 1) to create
970 2) to display.
971 3) to insert.
972 4) to delete.
973 5) to sort by price.
974 6) to search.
975 7) to modify.
976 8) create using queue.
977 9) display using queue.
978 10) delete using queue.
979 11) Create tree of price.
980 12) exit(0).
981
982 Enter your choice-11
983
984      2424
985
986
987 23
988 */
989

```