```
ASSIGNMENT 8
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#define COUNT 10

struct car_rent
{
  char car_name[20];
  char location[20];
  int price;
  int km;
  int time;
  struct car_rent *link;
}*start;

struct priceTree
{
  int price;
  struct priceTree *left;
  struct priceTree *right;
};
struct car_rent* getNewNode();
struct car_rent *create(struct car_rent *start);
struct car_rent *insert(struct car_rent *start,int n);
struct car_rent *del(struct car_rent *start,char item[]);
struct car_rent *sort (struct car_rent *start);
struct car_rent *modify(struct car_rent *start,char item[]);
void display (struct car_rent *start);
void *search(struct car_rent *start,char item[]);
void create_queue();
void display_queue();
void delete_queue();
struct car_rent *front=NULL;
struct car_rent *rear=NULL;
struct priceTree *node;
struct priceTree *root;
struct priceTree *getTree();
struct priceTree *getnode(int x);
struct priceTree *BalancedBst(int arr[],int s,int e);
void sortTree(int arr[],int n);
void print2DUtil(struct priceTree *root, int space);
void print2D(struct priceTree *root);
struct car_rent *p,*q,*temp;
char item[20];
int i,n,x,j,ch,l;
int s,e,point=0;
int flag=0;
void main()
{
   struct car_rent *start=NULL;

   printf("\t\t***************************************\n");
   printf("\t\t***************************************\n");
   printf("\t\t**   CAR RENTAL SYSTEM   **\n");
   printf("\t\t***************************************\n");
   printf("\t\t***************************************\n");
  while(1)
    {
     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 ");
     printf("Enter your choice-");
     scanf("%d",&ch);
      switch (ch)
      {
        case 1:
            start=create(start);
            break;
        case 2:
            display(start);
            break;
        case 3:
            start=insert(start,n);
            n++;
            break;
        case 4:
            printf("Enter item you want to delete:");
            scanf("%s",item);
            start=del(start,item);
            break;
        case 5:
            sort(start);
            break;
```

```
         case 6:
             printf("Enter item you want to search:");
             scanf("%s",item);
             search(start,item);
             break;
         case 7:
             printf("Enter item you want to modify:");
             scanf("%s",item);
             start=modify(start,item);
             break;
         case 8:
             create_queue();
             break;
         case 9:
             display_queue();
             break;
         case 10:
             delete_queue();
             break;
         case 11:
             print2D(getTree(start));
             break;
         case 12:
             exit(0);
             break;
         default:
             printf("error!");
     }

   }
}

struct car_rent *create(struct car_rent *start)
{

    struct car_rent *temp;
    printf("Enter no of cars you want to rent:");
    scanf("%d",&n);
    temp=start;
    for(i=0;i<n;i++)
     {
       temp=(struct car_rent*)malloc(sizeof(struct car_rent)*1);
       while(1)
       {
        printf("\tenter car name:");
        scanf("%s",(temp->car_name));
        printf("\tenter location");
        scanf("%s",(temp->location));
        l=strlen(temp->car_name);
        l=strlen(temp->location);
        if(l > 10 )
        {
          printf("\tPlease enter less characters in list name\n");
          printf("\tPlease enter less characters in list name\n");
        }
        else
        break;
       }

       do{
           printf("\tenter price:");
           scanf("%d",&(temp->price));
          if(temp->price > 1 && temp->price < 99999)
          {
            break;
          }
          else
          {
            printf("\tPlease enter less price\n");
          }
         }while(1);

       do{
          printf("\tenter time:");
          scanf("%d",&(temp->time));
            if(temp->time > 1 && temp->time < 55500)
          {
            break;
          }
          else
          {
            printf("\tPlease enter less time\n");
          }
        }while(1);
```

```
167
168         do{
169
170             printf("\tenter distance:");
171             scanf("%d",&(temp->km));
172             if(temp-> km > 0 && temp-> km < 22200)
173            {
174              break;
175            }
176            else
177            {
178              printf("\tSORRY..not available Please enter distance between(1 to 11111200)\n");
179            }
180          }while(1);
181         printf("\n");
182
183        if(start==NULL)
184        {
185         start=temp;
186         temp->link=NULL;
187        }
188
189        else
190        {
191            p=start;
192            while(p->link!=NULL)
193            p=p->link;
194            p->link=temp;
195            temp->link=NULL;
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("\tenter car name:");
237           scanf("%s",(temp->car_name));
238           printf("\tenter 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("\tenter 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("\tenter 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("\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
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("\tenter car name:");
309           scanf("%s",(temp->car_name));
310           printf("\tenter 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("\tenter 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("\tenter 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("\tenter 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\ttime\tdistance\n.");
429              printf("\n%s\t %d\t %s\t %d\t
      %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\ttime\tdistance");
439              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);
440             }
441              p=p->link;
442              flag++;
443           }
444
445           if(flag==0)
446           printf("\n\nnot found\n\n");
447
448   }
449
450   struct car_rent *modify(struct car_rent *start,char item[])
451   {
452       char ans1,ans2;
453       int chw=0;
454       if(strcmp(start->car_name,item)==0)
455        {
456          printf("\nITEM FOUND\n");
457          printf("\ncar name\tPrice\tlocation\ttime\tdistance\n.");
458          printf("\n%s\t %d\t %s\t %d\t
      %d\n",start->car_name,start->price,start->location,start->time,start->km);
459
460         do
461         {
462          printf("enter:\n 1) to modify car name\n 2) to modify list price:");
463          scanf("%d",&ch);
464          switch (ch)
465          {
466          case 1:
467                while(1)
468                {
469                 printf("\tenter product name:");
470                 scanf("%s",(temp->car_name));
471                 l=strlen(temp->car_name);
472                 if(l > 10 )
473                 {
474                   printf("\tPlease enter less characters in list name\n");
475                 }
476                 else
477                 {
478                    break;
479                 }
480                }
481                 break;
482
483          case 2:
484                 do{
485                    printf("\tenter price:");
486                    scanf("%d",&(temp->price));
487                   if(temp->price > 1 && temp->price < 99999)
488                   {
489                     break;
490                   }
491                   else
492                   {
493                     printf("\tPlease enter less price\n");
494                   }
495                 }while(1);
496                 break;
497
498          }
499
```

```
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("\tenter 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("\tenter 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("\tenter car name:");
589           scanf("%s",(temp->car_name));
590           printf("\tenter 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("\tenter 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("\tenter 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("\tenter 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 11111200)\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
```

```
void print2D(struct priceTree *root)
{
    print2DUtil(root, 0);
}









/*

OUTPUT
                ****************************************
                ****************************************
                **   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

```

```
        car name        Price   location        time    distance.

        audi    23      pune    34              64
        bmw     4225    mumbai  43              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-3
Enter location you want to insert:1
2
        enter car name:maruti
        enter locationsonari
        enter price:2424
        enter time:24
        enter distance :522
 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

        car name        Price   location        time    distance.

        maruti  2424    sonari  24              522
        audi    23      pune    34              64
        bmw     4225    mumbai  43              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-4
Enter item you want to delete:bmw
 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

        car name        Price   location        time    distance.

        maruti  2424    sonari  24              522
        audi    23      pune    34              64

 Enter your choice-5
 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

        car name        Price   location        time    distance.

        audi     23     pune    34              64
        maruti   2424   sonari  24              522
 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-6
Enter item you want to search:audi

ITEM FOUND

car name        Price   location        time    distance
.
audi     23      pune    34      64
 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).

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-11

          2424


23
*/
```