

```
3936 //Tickets
3937 #include<stdio.h>
3938 #include<stdlib.h>
3939 #define q_s 10
3940 int item,f=0,r=-1,q[q_s],c=0;
3941 void insert()
3942 {
3943     if(c==q_s) {
3944         printf("All tickets booked\n");
3945         return;
3946     }
3947     r=(r+1)%q_s;
3948     q[r]=item;
3949     c++;
3950 }
3951 int delete_front()
3952 {
3953     if(c==0) {
3954         return -1;
3955     }
3956     item=q[f];
3957     f=(f+1)%q_s;
3958     c=c-1;
3959     return item;
3960 }
3961 void display()
3962 {
3963     int i,front;
3964     if(c==0) {
3965         printf("No ticket booked\n");
3966         return;
3967     }
3968     front=f;
3969     printf("Ticket id : \n");
3970     for(i=0;i<c;i++) {
```

```
3969     printf("Ticket id : \n");
3970     for(i=0;i<c;i++) {
3971         printf("%d\n",q[front]);
3972         front=(front+1)%q_s;
3973     }
3974 }
3975 int main()
3976 {
3977     int ch;
3978     for(;;) {
3979         printf("\n1.Book a Ticket\n2.Cancel a Ticket\n3.display\n4.exit\n");
3980         printf("Enter the choice : ");
3981         scanf("%d", &ch);
3982         switch(ch) {
3983             case 1:printf("Enter the id of ticket : ");
3984                 scanf("%d", &item);
3985                 insert();
3986                 break;
3987             case 2:item=delete_front();
3988                 if(item==-1) {
3989                     printf("No ticket is booked yet\n");
3990                 }
3991                 else{
3992                     printf("Ticket cancelled id : %d\n", item);
3993                 }
3994                 break;
3995             case 3:display();
3996                 break;
3997             case 4:exit(0);
3998             default:printf("Enter proper choice!!!\n");
3999                 break;
4000             }
4001         }
4002     return 0;
4003 }
```

1.Book a Ticket
2.Cancel a Ticket
3.display
4.exit

Enter the choice : 1
Enter the id of ticket : 45

1.Book a Ticket
2.Cancel a Ticket
3.display
4.exit

Enter the choice : 1
Enter the id of ticket : 98

1.Book a Ticket
2.Cancel a Ticket
3.display
4.exit

Enter the choice : 1
Enter the id of ticket : 74

1.Book a Ticket
2.Cancel a Ticket
3.display
4.exit

Enter the choice : 1
Enter the id of ticket : 32

1.Book a Ticket
2.Cancel a Ticket
3.display
4.exit

Enter the choice : 3
Ticket id :
45
98
74
32

1.Book a Ticket
2.Cancel a Ticket
3.display
4.exit

Enter the choice : 2
Ticket cancelled id : 45

```
4008 //Employee more than 80 hours
4009 #include <stdio.h>
4010 #include <stdlib.h>
4011 void insertAtBeg();
4012 void insertAtEnd();
4013 void insertAtAfter();
4014 void display();
4015 struct emp{
4016     char name[20];
4017     int data,m,h;
4018 };
4019 struct node
4020 {
4021     struct emp e;
4022     struct node *next;
4023 };
4024 struct node *head=NULL;
4025 int main(int argc, char **argv)
4026 {
4027     int choice;
4028     char ch='y';
4029     do
4030     {
4031         printf("\n1. InsertAtEnd \n2.InsertAtBeg \n3. InsertAtAfter \n4. Display\n5. Exit\n ");
4032         printf("\nEnter your choice : ");
4033         scanf("%d", &choice);
4034         switch(choice)
4035         {
4036             case 1: insertAtEnd(); break;
4037             case 2: insertAtBeg(); break;
4038             case 3: insertAtAfter(); break;
4039             case 4: display();break;
4040             case 5: ch='n'; break;
4041         }
4042     }while(ch=='y' || ch=='Y');
```

```
4041     }
4042     }while(ch=='y' || ch=='Y');
4043 }
4044
4045 void insertAtEnd()
4046 {
4047     struct node *newnode,*temp;
4048     newnode =(struct node *) malloc (sizeof(struct node));
4049     printf("Enter the name : ");
4050     scanf("%s",newnode->e.name);
4051     printf("Enter the id : ");
4052     scanf("%d",&newnode->e.data);
4053     printf("Enter the mobile : ");
4054     scanf("%d",&newnode->e.m);
4055     printf("Enter the no. of hours : ");
4056     scanf("%d",&newnode->e.h);
4057     newnode->next=NULL;
4058     if (head==NULL)
4059     {
4060         head=newnode;
4061         newnode->next=newnode;
4062
4063         printf("Information stored\n");
4064     }
4065     else
4066     {
4067         temp=head;
4068         while(temp->next!=head)
4069         {
4070             temp=temp->next;
4071         }
4072         temp->next=newnode;
4073         newnode->next=head;
4074         printf("Information stored\n");
4075     }
}
```

```
4074         printf("Information stored\n");
4075     }
4076 }
4077 void insertAtBeg()
4078 {
4079     struct node *newnode,*temp;
4080     newnode =(struct node *) malloc (sizeof(struct node));
4081     printf("Enter the name : ");
4082     scanf("%s",newnode->e.name);
4083     printf("Enter the id : ");
4084     scanf("%d",&newnode->e.data);
4085     printf("Enter the mobile : ");
4086     scanf("%d",&newnode->e.m);
4087     printf("Enter the no. of hours : ");
4088     scanf("%d",&newnode->e.h);
4089     newnode->next=NULL;
4090     if (head==NULL)
4091     {
4092         head=newnode;
4093         newnode->next=newnode;
4094
4095         printf("Information stored\n");
4096     }
4097     else
4098     {
4099         temp=head;
4100         while(temp->next!=head)
4101         {
4102             temp=temp->next;
4103         }
4104         newnode->next=head;
4105         head=newnode;
4106         temp->next=head;
4107         printf("Information stored\n");
4108     }
```

```
4107         printf("Information stored\n");
4108     }
4109 }
4110 void insertAtAfter()
4111 {
4112     struct node *newnode,*temp;
4113     int ele, flag=0;
4114     if (head==NULL)
4115     {
4116         printf("Empty list\n");
4117     }
4118     else
4119     {
4120         newnode =(struct node *) malloc (sizeof(struct node));
4121         printf("Enter the name : ");
4122         scanf("%s",newnode->e.name);
4123         printf("Enter the id : ");
4124         scanf("%d",&newnode->e.data);
4125         printf("Enter the mobile : ");
4126         scanf("%d",&newnode->e.m);
4127         printf("Enter the no. of hours : ");
4128         scanf("%d",&newnode->e.h);
4129         newnode->next=NULL;
4130         printf("Enter the employee id after which insertion to be done\n");
4131         scanf("%d",&ele);
4132         temp=head;
4133         while(temp->next!=head)
4134         {
4135             if(temp->e.data==ele)
4136             {
4137                 newnode->next=temp->next;
4138                 temp->next=newnode;
4139                 flag=1;break;
4140             }
4141             temp=temp->next;
```

```
4140         }
4141         temp=temp->next;
4142     }
4143     if((temp->e.data==ele) && (flag==0))
4144     {
4145         newnode->next=temp->next;
4146         temp->next=newnode;
4147         flag=1;
4148     }
4149
4150     if (flag == 0)
4151     {
4152         printf("\n Employee id not found\n");
4153     }
4154 }
4155 }
4156 void display()
4157 {
4158     struct node *ptr=NULL;
4159     ptr=head;
4160     int j=0;
4161
4162     if(ptr==NULL)
4163     {
4164         printf("Empty list\n");
4165     }
4166     else
4167     {
4168         do
4169         {
4170             if(ptr->e.h>80)
4171             j++;
4172             printf("\nName : %s\n",ptr->e.name);
4173             printf("Employee id : %d\n",ptr->e.data);
4174             printf("Mobile No. : %d\n",ptr->e.m);
```

```
4163 }  
4164 else  
4165 {  
4166     do  
4167     {  
4168         if(ptr->e.h>80){  
4169             j++;  
4170             printf("\nName : %s\n",ptr->e.name);  
4171             printf("Employee id : %d\n",ptr->e.data);  
4172             printf("Mobile No. : %d\n",ptr->e.m);  
4173             printf("No. of hours worked : %d\n",ptr->e.h);  
4174         }  
4175         ptr=ptr->next;  
4176     }while(ptr!=head);  
4177 }  
4178 }  
4179
```

1. InsertAtEnd
2. InsertAtBeg
3. InsertAtAfter
4. Display
5. Exit

Enter your choice : 1

Enter the name : a

Enter the id : 14

Enter the mobile : 8745

Enter the no. of hours : 120

Information stored

1. InsertAtEnd
2. InsertAtBeg
3. InsertAtAfter
4. Display
5. Exit

Enter your choice : 2

Enter the name : b

Enter the id : 456

Enter the mobile : 9632

Enter the no. of hours : 79

Information stored

1. InsertAtEnd
2. InsertAtBeg
3. InsertAtAfter
4. Display
5. Exit

Enter your choice : 4

Name : a

Employee id : 14

Mobile No. : 8745

No. of hours worked : 120

1. InsertAtEnd
2. InsertAtBeg
3. InsertAtAfter
4. Display
5. Exit

Enter your choice : 5

Process returned 0 (0x0) execution time : 54.419 s

Press any key to continue.

```
4179 //Resides in Bangalore
4180 #include <stdio.h>
4181 #include <stdlib.h>
4182 #include <string.h>
4183 void insertAtBeg();
4184 void insertAtEnd();
4185 void insertAtAfter();
4186 void display();
4187 struct emp{
4188     char name[20],p[20];
4189     int data,m;
4190 };
4191 struct node
4192 {
4193     struct emp e;
4194     struct node *next;
4195 };
4196 struct node *head=NULL;
4197 int main(int argc, char **argv)
4198 {
4199     int choice;
4200     char ch='y';
4201     do
4202     {
4203         printf("\n1. InsertAtEnd \n2.InsertAtBeg \n3. InsertAtAfter \n4. Display\n5. Exit\n ");
4204         printf("\nEnter your choice : ");
4205         scanf("%d",&choice);
4206         switch(choice)
4207         {
4208             case 1: insertAtEnd(); break;
4209             case 2: insertAtBeg(); break;
4210             case 3: insertAtAfter(); break;
4211             case 4: display();break;
4212             case 5: ch='n'; break;
4213         }
4214     }while(ch=='y');
4215 }
```

```
4211         case 4: display();break;
4212         case 5: ch='n'; break;
4213     }
4214 }while(ch=='y' || ch=='Y');
4215
4216
4217 void insertAtEnd()
4218 {
4219     struct node *newnode,*temp;
4220     newnode =(struct node *) malloc (sizeof(struct node));
4221     printf("Enter the name : ");
4222     scanf("%s",newnode->e.name);
4223     printf("Enter the id : ");
4224     scanf("%d",&newnode->e.data);
4225     printf("Enter the mobile : ");
4226     scanf("%d",&newnode->e.m);
4227     printf("Enter the place : ");
4228     scanf("%s",newnode->e.p);
4229     newnode->next=NULL;
4230     if (head==NULL)
4231     {
4232         head=newnode;
4233         newnode->next=newnode;
4234
4235         printf("Information stored\n");
4236     }
4237     else
4238     {
4239         temp=head;
4240         while(temp->next!=head)
4241         {
4242             temp=temp->next;
4243         }
4244         temp->next=newnode;
4245         newnode->next=head;
```

```
4244     temp->next=newnode;
4245     newnode->next=head;
4246     printf("Information stored\n");
4247 }
4248 }
4249 void insertAtBeg()
4250 {
4251     struct node *newnode,*temp;
4252     newnode =(struct node *) malloc (sizeof(struct node));
4253     printf("Enter the name : ");
4254     scanf("%s",newnode->e.name);
4255     printf("Enter the id : ");
4256     scanf("%d",&newnode->e.data);
4257     printf("Enter the mobile : ");
4258     scanf("%d",&newnode->e.m);
4259     printf("Enter the place : ");
4260     scanf("%s",newnode->e.p);
4261     newnode->next=NULL;
4262     if (head==NULL)
4263     {
4264         head=newnode;
4265         newnode->next=newnode;
4266
4267         printf("Information stored\n");
4268     }
4269     else
4270     {
4271         temp=head;
4272         while(temp->next!=head)
4273         {
4274             temp=temp->next;
4275         }
4276         newnode->next=head;
4277         head=newnode;
4278         temp->next=head;
```

```
4274         temp=temp->next;
4275     }
4276     newnode->next=head;
4277     head=newnode;
4278     temp->next=head;
4279     printf("Information stored\n");
4280 }
4281 }
4282 void insertAtAfter()
4283 {
4284     struct node *newnode, *temp;
4285     int ele, flag=0;
4286     if (head==NULL)
4287     {
4288         printf("Empty list\n");
4289     }
4290     else
4291     {
4292         newnode =(struct node *) malloc (sizeof(struct node));
4293         printf("Enter the name : ");
4294         scanf("%s", newnode->e.name);
4295         printf("Enter the id : ");
4296         scanf("%d", &newnode->e.data);
4297         printf("Enter the mobile : ");
4298         scanf("%d", &newnode->e.m);
4299         printf("Enter the place : ");
4300         scanf("%s", newnode->e.p);
4301         newnode->next=NULL;
4302         printf("Enter the employee id after which insertion to be done\n");
4303         scanf("%d", &ele);
4304         temp=head;
4305         while(temp->next!=head)
4306         {
4307             if(temp->e.data==ele)
4308             {
```

```
4307         if(temp->e.data==ele)
4308     {
4309         newnode->next=temp->next;
4310         temp->next=newnode;
4311         flag=1;break;
4312     }
4313     temp=temp->next;
4314 }
4315 if((temp->e.data==ele) && (flag==0))
4316 {
4317     newnode->next=temp->next;
4318     temp->next=newnode;
4319     flag=1;
4320 }
4321
4322 if(flag == 0)
4323 {
4324     printf("\n Employee id not found\n");
4325 }
4326 }
4327 }
4328 void display()
4329 {
4330     struct node *ptr=NULL;
4331     ptr=head;
4332     int j=0,value;
4333     char s7[]="Bangalore";
4334     if(ptr==NULL)
4335     {
4336         printf("Empty list\n");
4337     }
4338     else
4339     {
4340         do
4341         {
```

```
4340     do
4341     {
4342         value=strcmp(ptr->e.p,s7);
4343         if(value==0){
4344             j++;
4345             printf("\nName : %s\n",ptr->e.name);
4346             printf("Place : %s\n",ptr->e.p);
4347             printf("Mobile No. : %d\n",ptr->e.m);
4348         }
4349         ptr=ptr->next;
4350     }while(ptr!=head);
4351 }
4352 }*/
```

1. InsertAtEnd
2. InsertAtBeg
3. InsertAtAfter
4. Display
5. Exit

Enter your choice : 1

Enter the name : a

Enter the id : 12

Enter the mobile : 1234

Enter the place : Bangalore

Information stored

1. InsertAtEnd
2. InsertAtBeg
3. InsertAtAfter
4. Display
5. Exit

Enter your choice : 3

Enter the name : b

Enter the id : 76

Enter the mobile : 4321

Enter the place : Mysore

Enter the employee id after which insertion to be done

12

1. InsertAtEnd
2. InsertAtBeg
3. InsertAtAfter
4. Display
5. Exit

Enter your choice : 4

Name : a

Place : Bangalore

Mobile No. : 1234

1. InsertAtEnd
2. InsertAtBeg
3. InsertAtAfter
4. Display
5. Exit

Enter your choice : 5

```
4358 //Back button
4359 #include<stdio.h>
4360 #include<stdlib.h>
4361 struct node
4362 {
4363     int info;
4364     struct node *rlink;
4365     struct node *llink;
4366 };
4367 typedef struct node *NODE;
4368 NODE getnode()
4369 {
4370     NODE x;
4371     x=(NODE)malloc(sizeof(struct node));
4372     if(x==NULL)
4373         printf("Memory full\n");
4374         exit(0);
4375     }
4376     return x;
4377 }
4378 void freenode(NODE x)
4379 {
4380     free(x);
4381 }
4382 NODE dinsert_front(NODE head,int item)
4383 {
4384     NODE temp,cur;
4385     temp=getnode();
4386     temp->rlink=NULL;
4387     temp->llink=NULL;
4388     temp->info=item;
4389     cur=head->rlink;
4390     temp->rlink=cur;
4391     cur->llink=temp;
4392     head->rlink=temp;
```

```
4391     cur->llink=temp;
4392     head->rlink=temp;
4393     temp->llink=head;
4394     head->info=head->info+1;
4395     return head;
4396 }
4397 NODE ddelete_front(NODE head)
4398 {
4399     NODE cur,next;
4400     if(head->rlink==head) {
4401         printf("No url entered\n");
4402         return head;
4403     }
4404     cur=head->rlink;
4405     next=cur->rlink;
4406     head->rlink=next;
4407     next->llink=head;
4408     printf("Deleted url id is %d\n",cur->info);
4409     freenode(cur);
4410     return head;
4411 }
4412 void display(NODE head)
4413 {
4414     NODE temp;
4415     if(head->rlink==head) {
4416         printf("No url entered\n");
4417         return;
4418     }
4419     for(temp=head->rlink,temp!=head,temp=temp->rlink) {
4420         printf("%d\n",temp->info);
4421     }
4422 }
4423 int main()
4424 {
4425     int item,choice;
```

```
4421 }
4422 }
4423 int main()
4424 {
4425     int item,choice;
4426     NODE head,temp;
4427     head=getnode();
4428     head->rlink=head;
4429     head->llink=head;
4430     for(;;){
4431         printf("\n1.Visit new url\n2.Back\n3.Display\n4.Exit\n");
4432         printf("enter the choice\n");
4433         scanf("%d",&choice);
4434         switch(choice){
4435             case 1:printf("Enter the new url id :- ");
4436                 scanf("%d",&item);
4437                 head=dinsert_front(head,item);
4438                 break;
4439             case 2:head=ddelete_front(head);
4440                 temp=head->rlink;
4441                 break;
4442             case 3:printf("Url's:\n");
4443                 display(head);
4444                 break;
4445             case 4:exit(0);
4446                 break;
4447             default:printf("Enter proper instructions!!!\n");
4448                 break;
4449         }
4450     }
4451     return 0;
4452 }
4453 }
```

1.Visit new url
2.Back
3.Display
4.Exit
enter the choice

1
Enter the new url id :- 23

1.Visit new url
2.Back
3.Display
4.Exit
enter the choice

1
Enter the new url id :- 45

1.Visit new url
2.Back
3.Display
4.Exit
enter the choice

1
Enter the new url id :- 87

1.Visit new url
2.Back
3.Display
4.Exit
enter the choice

3
Url's:
87
45
23

1.Visit new url
2.Back
3.Display
4.Exit
enter the choice

2
Deleted url id is 87

1.Visit new url
2.Back
3.Display
4.Exit
enter the choice

3

["C:\Users\Shreshtha Aggarwal\Desktop\1stpro\bin\Debug\1stpro.exe"]

Url's:

45

23

1.Visit new url

2.Back

3.Display

4.Exit

enter the choice

4

Process returned 0 (0x0) execution time : 20.882 s

Press any key to continue.

```
4455 //Reverse name
4456 #include<stdio.h>
4457 #include<stdlib.h>
4458 struct name{
4459     char d[20];
4460 };
4461 struct node
4462 {
4463     struct name n;
4464     struct node *link;
4465 };
4466 typedef struct node *NODE;
4467 NODE getnode()
4468 {
4469     NODE x;
4470     x=(NODE)malloc(sizeof(struct node));
4471     if(x==NULL)
4472     {
4473         printf("memory full\n");
4474         exit(0);
4475     }
4476     return x;
4477 }
4478 void freenode(NODE x)
4479 {
4480     free(x);
4481 }
4482 NODE insert_r(NODE first)
4483 {
4484     NODE temp,cur;
4485     temp=getnode();
4486     printf("Enter the name : \n");
4487     scanf("%s",temp->n.d);
4488     temp->link=NULL;
```

```
4487     scanf ("%s", temp->n.d);
4488     temp->link=NULL;
4489     if(first==NULL)
4490         return temp;
4491     cur=first;
4492     while(cur->link!=NULL)
4493         cur=cur->link;
4494     cur->link=temp;
4495     return first;
4496 }
4497 NODE reverse(NODE first)
4498 {
4499     NODE cur,temp;
4500     cur=NULL;
4501     while(first!=NULL)
4502     {
4503         temp=first;
4504         first=first->link;
4505         temp->link=cur;
4506         cur=temp;
4507     }
4508     return cur;
4509 }
4510 void display(NODE first)
4511 {
4512     NODE temp;
4513     if(first==NULL)
4514         printf("Name not entered\n");
4515     for(temp=first;temp!=NULL;temp=temp->link)
4516     {
4517         printf("%s\n", temp->n.d);
4518     }
4519 }
4520 int main()
4521 {
```

```
4514         printf("Name not entered\n");
4515     for(temp=first,temp!=NULL,temp=temp->link)
4516     {
4517         printf("%s\n",temp->n.d);
4518     }
4519 }
4520 int main()
4521 {
4522     int choice;
4523     NODE first=NULL;
4524     for(;;){
4525         printf("\n1.InsertinReverse\n2.Display\n3.Exit\n");
4526         printf("Enter the choice\n");
4527         scanf("%d", &choice);
4528         switch(choice)
4529         {
4530             case 1:first=insert_r(first);
4531                 break;
4532             case 2:first=reverse(first);
4533                 printf("List : \n");
4534                 display(first);
4535                 break;
4536             case 3:exit(0);
4537             default:printf("Enter correct instruction!!!\n");
4538                 break;
4539         }
4540     }
4541     return 0;
4542 }
```

"C:\Users\Shreshtha Aggarwal\Desktop\1stpro\bin\Debug\1stpro.exe"

1:InsertinReverse
2.Display
3.Exit
Enter the choice
1
Enter the name :
f

1:InsertinReverse
2.Display
3.Exit
Enter the choice
1
Enter the name :
d

1:InsertinReverse
2.Display
3.Exit
Enter the choice
1
Enter the name :
s

1:InsertinReverse
2.Display
3.Exit
Enter the choice
1
Enter the name :
a

1:InsertinReverse
2.Display
3.Exit
Enter the choice
2
List :
a
s
d
f

1:InsertinReverse
2.Display
3.Exit
Enter the choice
3

Process returned 0 (0x0) execution time : 23.312 s