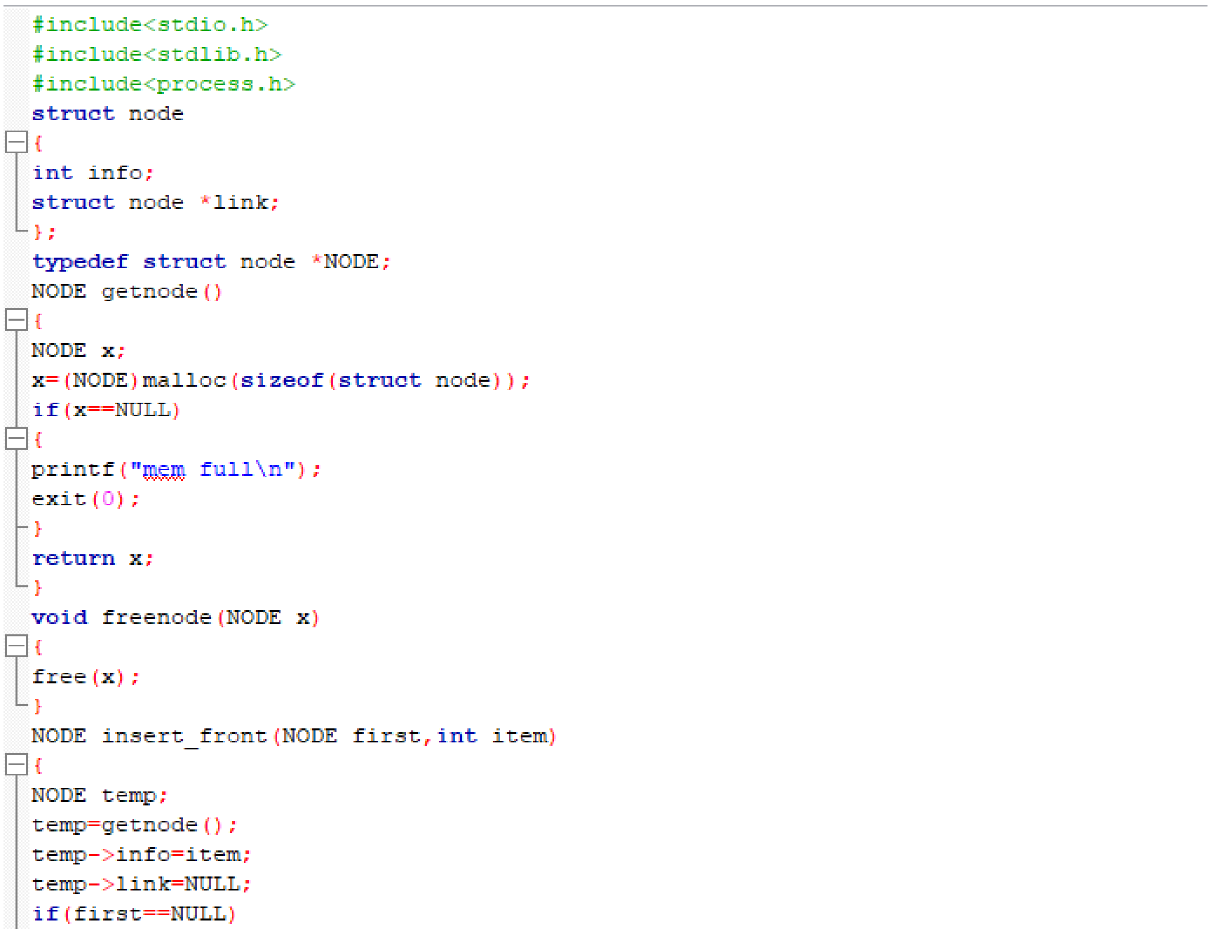
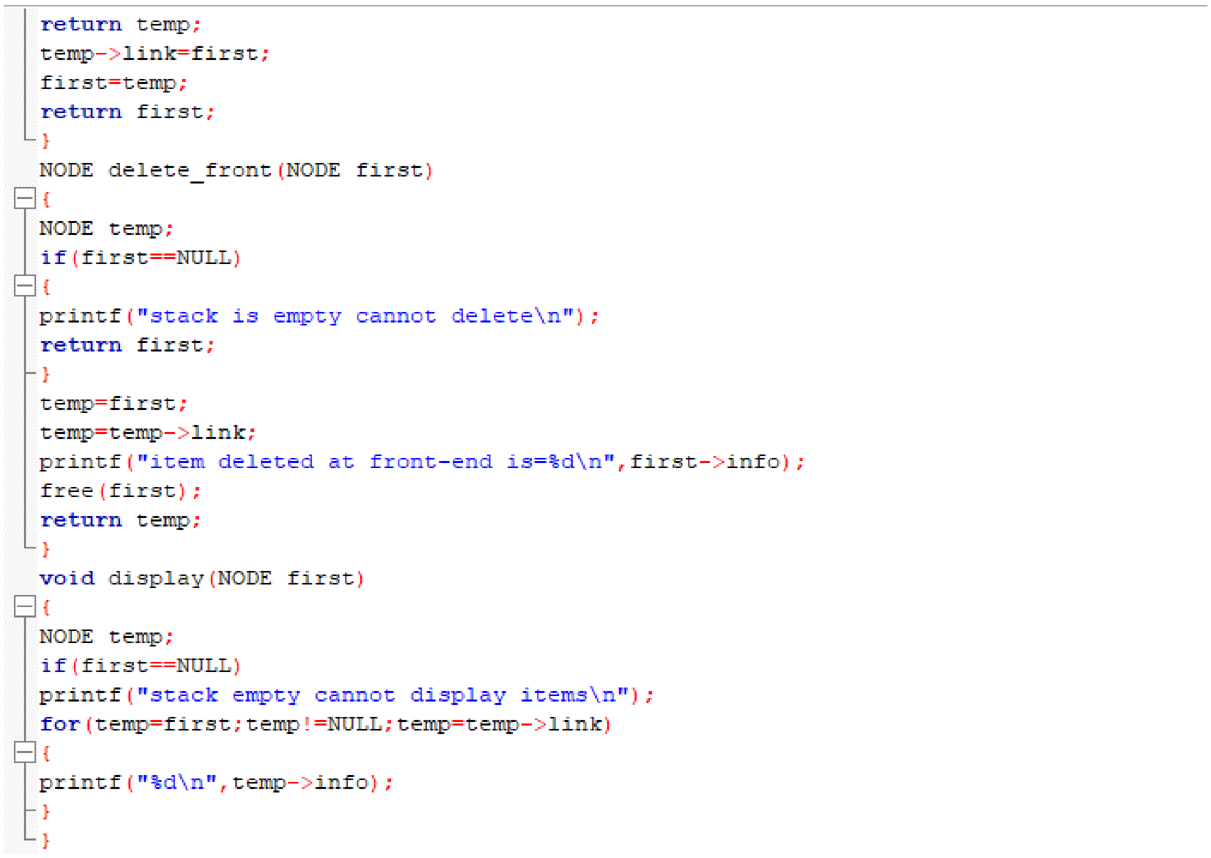
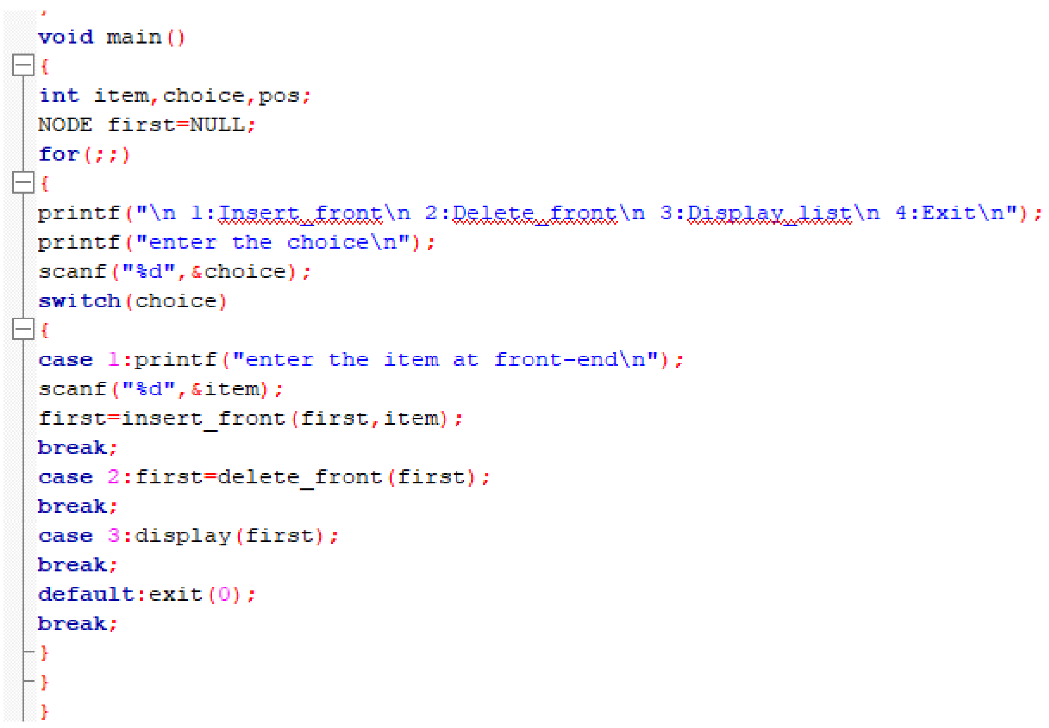


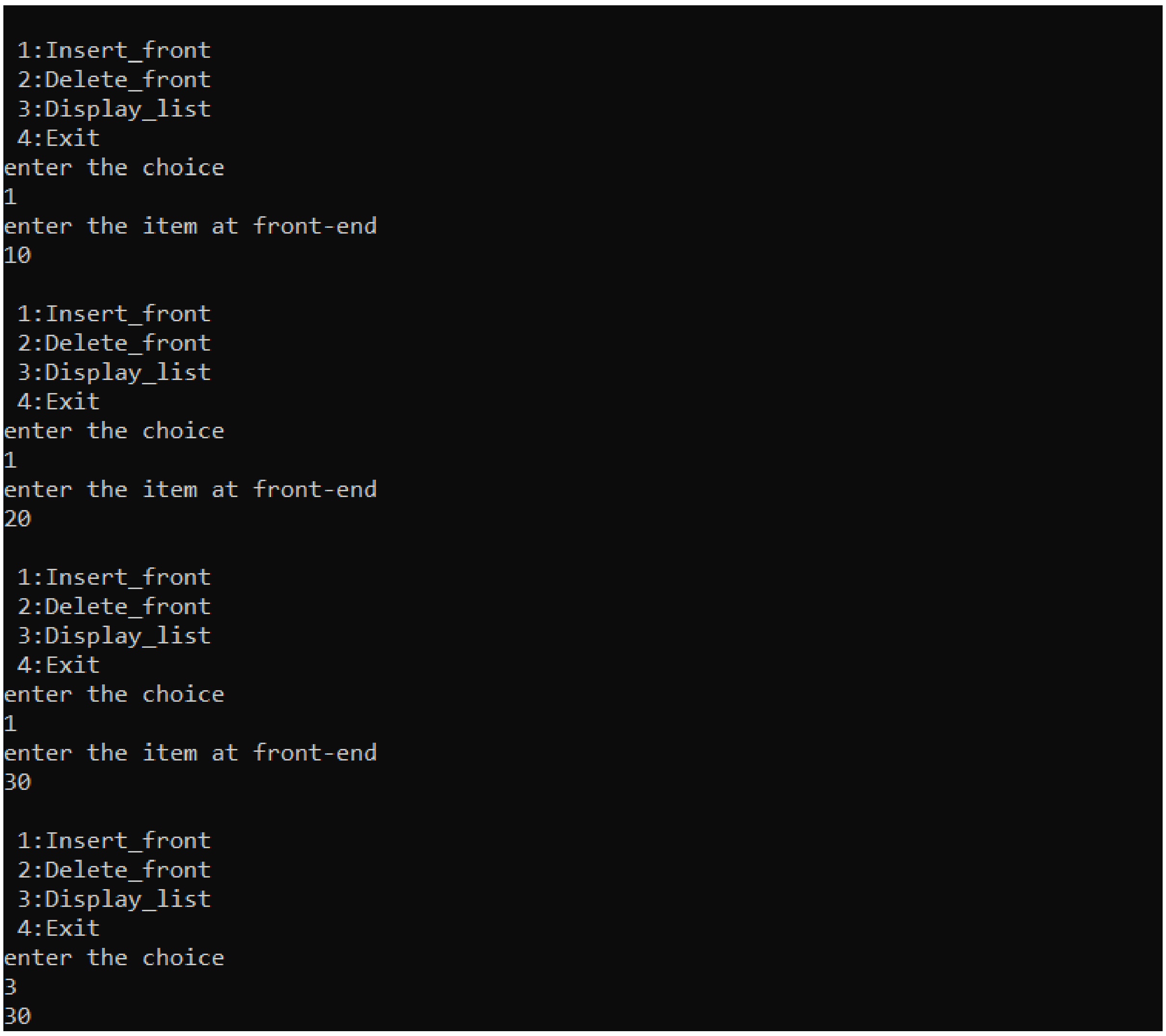
```c
#include<stdio.h>
#include<stdlib.h>
#include<process.h>
struct node
{
int info;
struct node *link;
};
typedef struct node *NODE;
NODE getnode()
{
NODE x;
x=(NODE)malloc(sizeof(struct node));
if(x==NULL)
{
printf("mem full\n");
exit(0);
}
return x;
}
void freenode(NODE x)
{
free(x);
}
NODE insert_front(NODE first,int item)
{
NODE temp;
temp=getnode();
temp->info=item;
temp->link=NULL;
if(first==NULL)
```



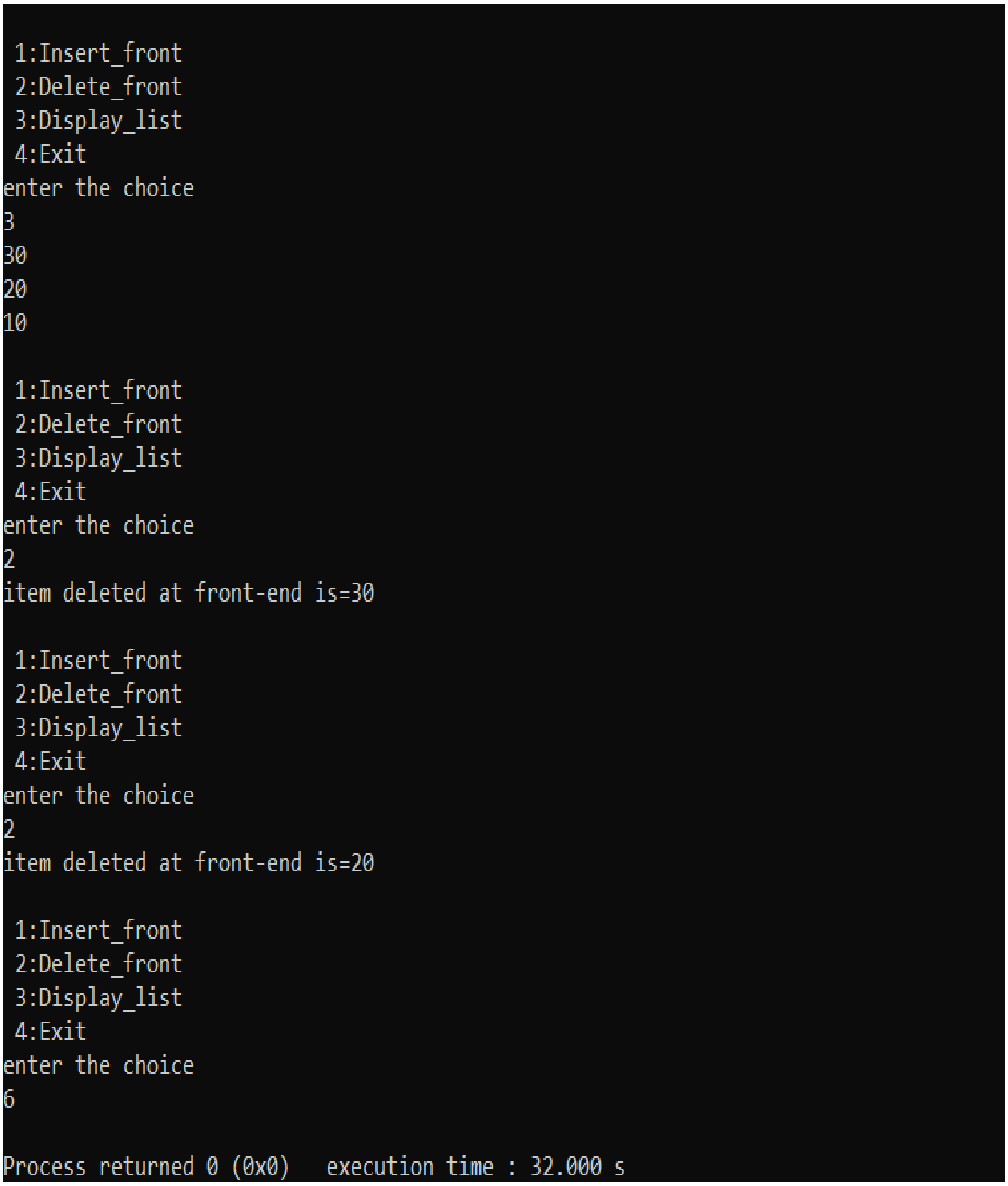
```c
return temp;
temp->link=first;
first=temp;
return first;
}
NODE delete_front(NODE first)
{
NODE temp;
if(first==NULL)
{
printf("stack is empty cannot delete\n");
return first;
}
temp=first;
temp=temp->link;
printf("item deleted at front-end is=%d\n",first->info);
free(first);
return temp;
}
void display(NODE first)
{
NODE temp;
if(first==NULL)
printf("stack empty cannot display items\n");
for(temp=first;temp!=NULL;temp=temp->link)
{
printf("%d\n",temp->info);
}
}
```



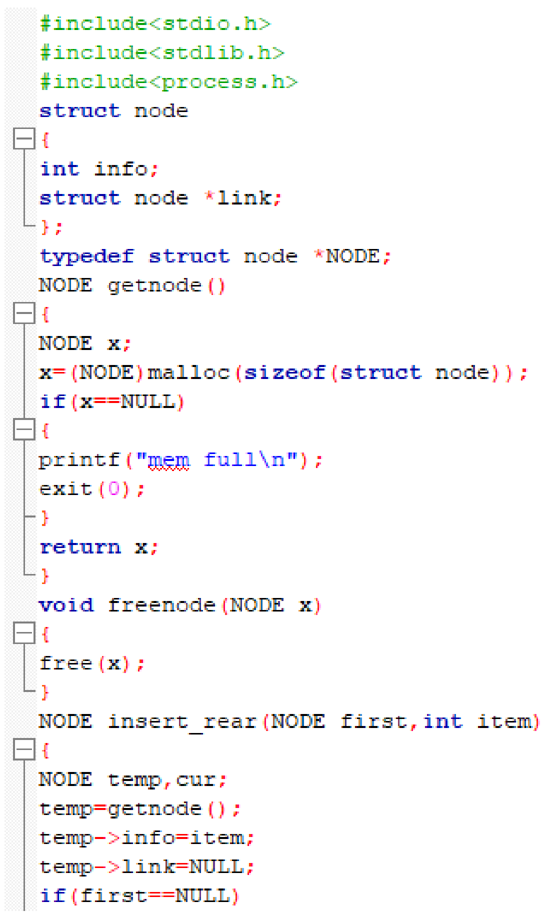
```c
void main()
{
int item,choice,pos;
NODE first=NULL;
for(;;)
{
printf("\n 1:Insert_front\n 2:Delete_front\n 3:Display_list\n 4:Exit\n");
printf("enter the choice\n");
scanf("%d",&choice);
switch(choice)
{
case 1:printf("enter the item at front-end\n");
scanf("%d",&item);
first=insert_front(first,item);
break;
case 2:first=delete_front(first);
break;
case 3:display(first);
break;
default:exit(0);
break;
}
}
}
```



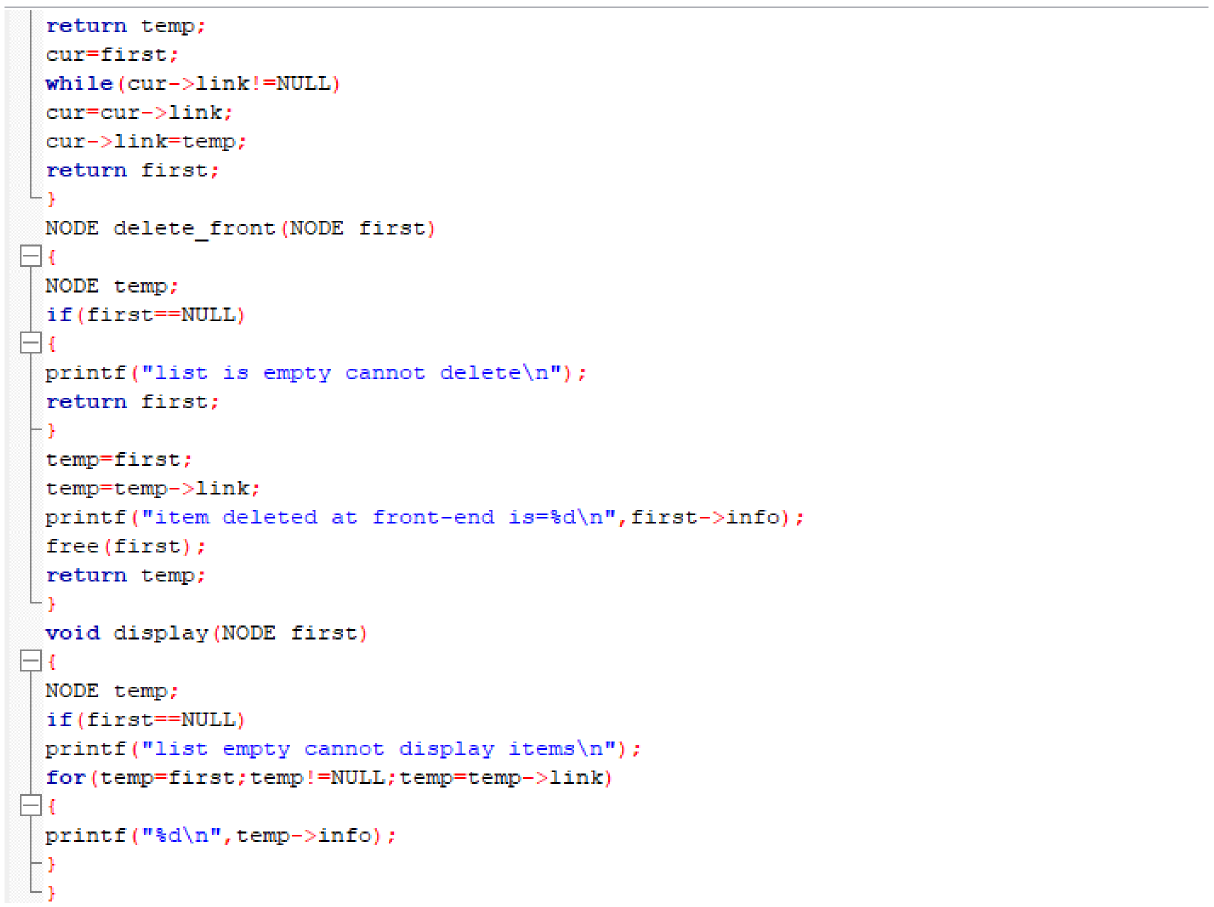
```
 1:Insert_front
 2:Delete_front
 3:Display_list
 4:Exit
enter the choice
1
enter the item at front-end
10

 1:Insert_front
 2:Delete_front
 3:Display_list
 4:Exit
enter the choice
1
enter the item at front-end
20

 1:Insert_front
 2:Delete_front
 3:Display_list
 4:Exit
enter the choice
1
enter the item at front-end
30

 1:Insert_front
 2:Delete_front
 3:Display_list
 4:Exit
enter the choice
3
30
```



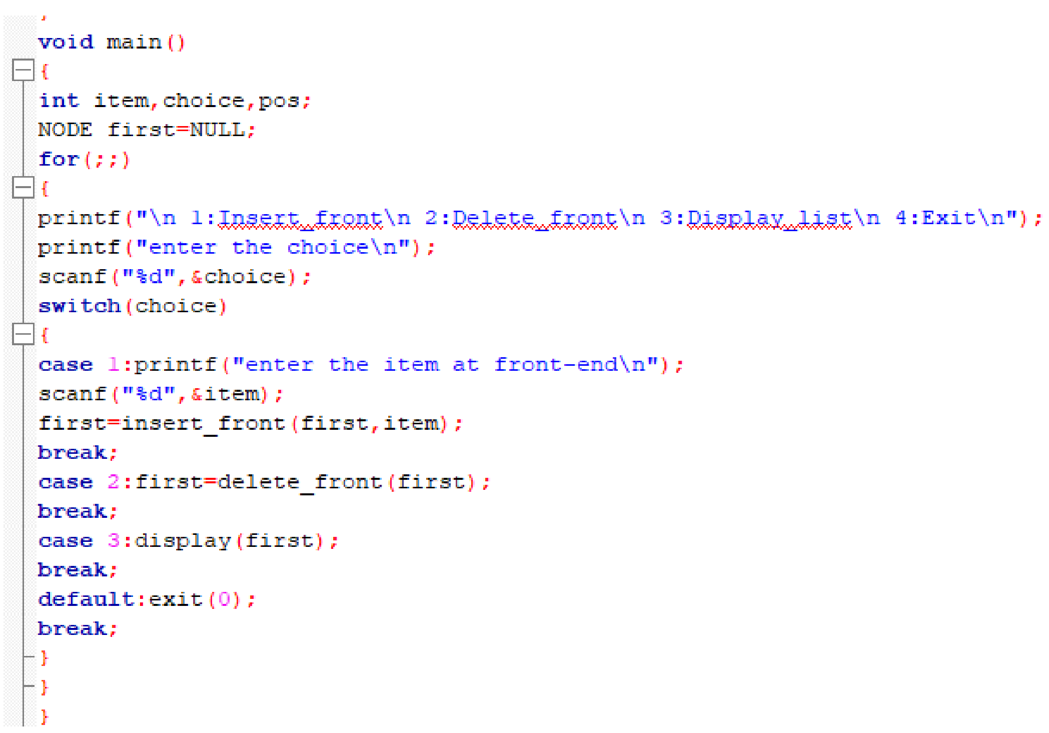
```

 1:Insert_front
 2:Delete_front
 3:Display_list
 4:Exit
enter the choice
3
30
20
10

 1:Insert_front
 2:Delete_front
 3:Display_list
 4:Exit
enter the choice
2
item deleted at front-end is=30

 1:Insert_front
 2:Delete_front
 3:Display_list
 4:Exit
enter the choice
2
item deleted at front-end is=20

 1:Insert_front
 2:Delete_front
 3:Display_list
 4:Exit
enter the choice
6

Process returned 0 (0x0)   execution time : 32.000 s
```



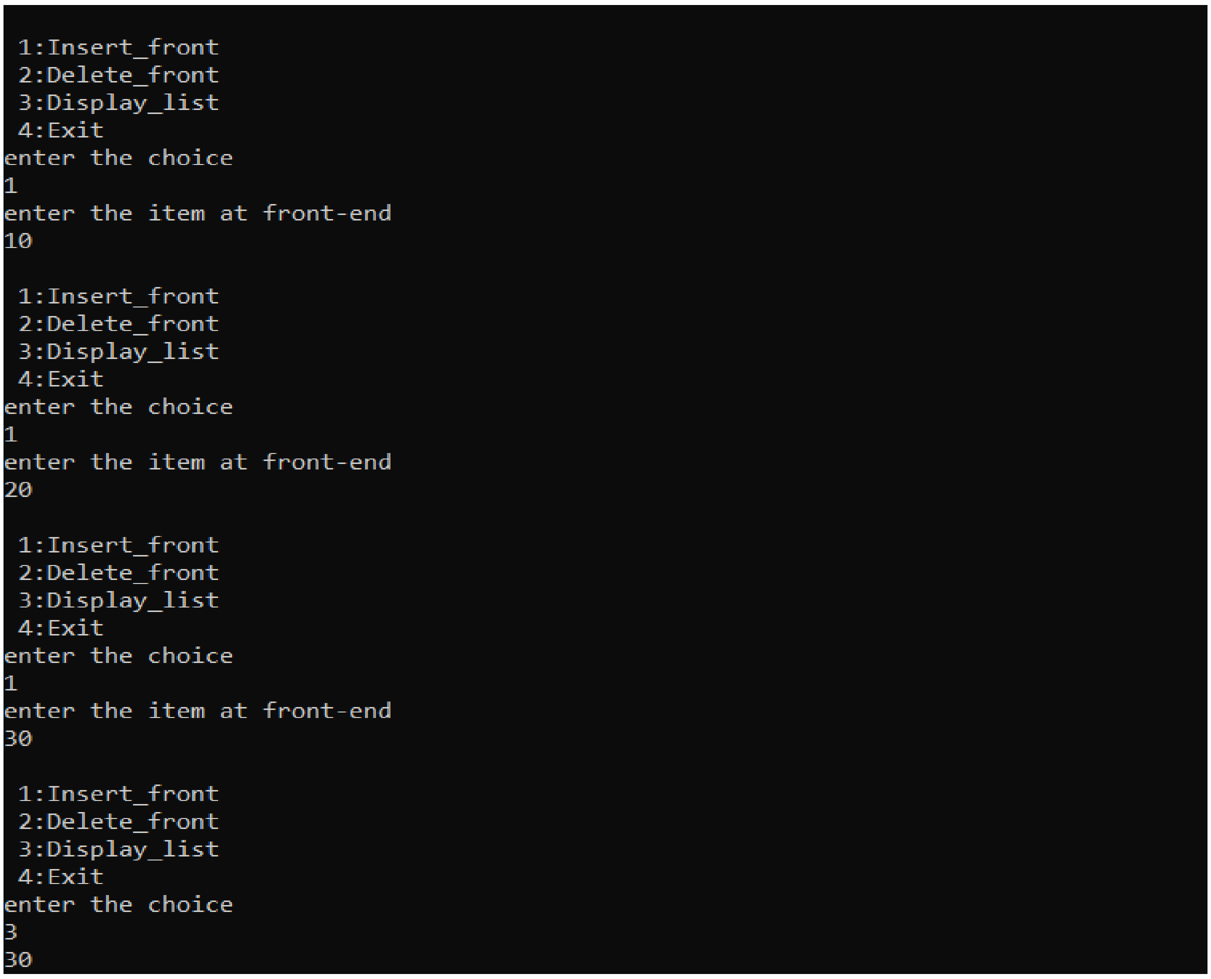
```
#include<stdio.h>
#include<stdlib.h>
#include<process.h>
struct node
{
int info;
struct node *link;
};
typedef struct node *NODE;
NODE getnode()
{
NODE x;
x=(NODE)malloc(sizeof(struct node));
if(x==NULL)
{
printf("mem full\n");
exit(0);
}
return x;
}
void freenode(NODE x)
{
free(x);
}
NODE insert_rear(NODE first,int item)
{
NODE temp,cur;
temp=getnode();
temp->info=item;
temp->link=NULL;
if(first==NULL)
```



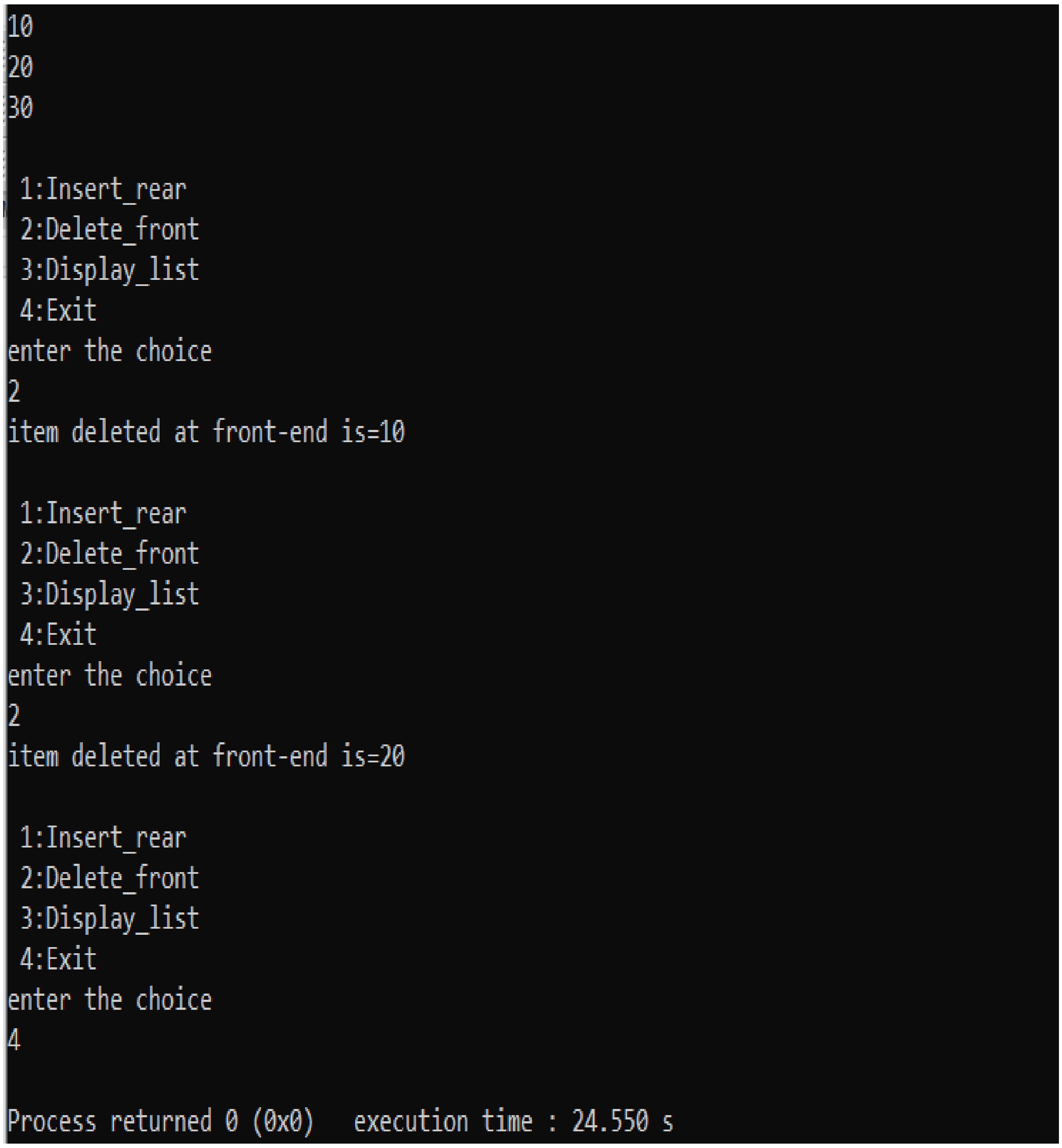
```
return temp;
cur=first;
while(cur->link!=NULL)
cur=cur->link;
cur->link=temp;
return first;
}
NODE delete_front(NODE first)
{
NODE temp;
if(first==NULL)
{
printf("list is empty cannot delete\n");
return first;
}
temp=first;
temp=temp->link;
printf("item deleted at front-end is=%d\n",first->info);
free(first);
return temp;
}
void display(NODE first)
{
NODE temp;
if(first==NULL)
printf("list empty cannot display items\n");
for(temp=first;temp!=NULL;temp=temp->link)
{
printf("%d\n",temp->info);
}
}
```



```
void main()
{
int item,choice,pos;
NODE first=NULL;
for(;;)
{
printf("\n 1:Insert_front\n 2:Delete_front\n 3:Display_list\n 4:Exit\n");
printf("enter the choice\n");
scanf("%d",&choice);
switch(choice)
{
case 1:printf("enter the item at front-end\n");
scanf("%d",&item);
first=insert_front(first,item);
break;
case 2:first=delete_front(first);
break;
case 3:display(first);
break;
default:exit(0);
break;
}
}
}
```



```
 1:Insert_front
 2:Delete_front
 3:Display_list
 4:Exit
enter the choice
1
enter the item at front-end
10

 1:Insert_front
 2:Delete_front
 3:Display_list
 4:Exit
enter the choice
1
enter the item at front-end
20

 1:Insert_front
 2:Delete_front
 3:Display_list
 4:Exit
enter the choice
1
enter the item at front-end
30

 1:Insert_front
 2:Delete_front
 3:Display_list
 4:Exit
enter the choice
3
30
```



```
10
20
30

 1:Insert_rear
 2:Delete_front
 3:Display_list
 4:Exit
enter the choice
2
item deleted at front-end is=10

 1:Insert_rear
 2:Delete_front
 3:Display_list
 4:Exit
enter the choice
2
item deleted at front-end is=20

 1:Insert_rear
 2:Delete_front
 3:Display_list
 4:Exit
enter the choice
4

Process returned 0 (0x0)   execution time : 24.550 s
```