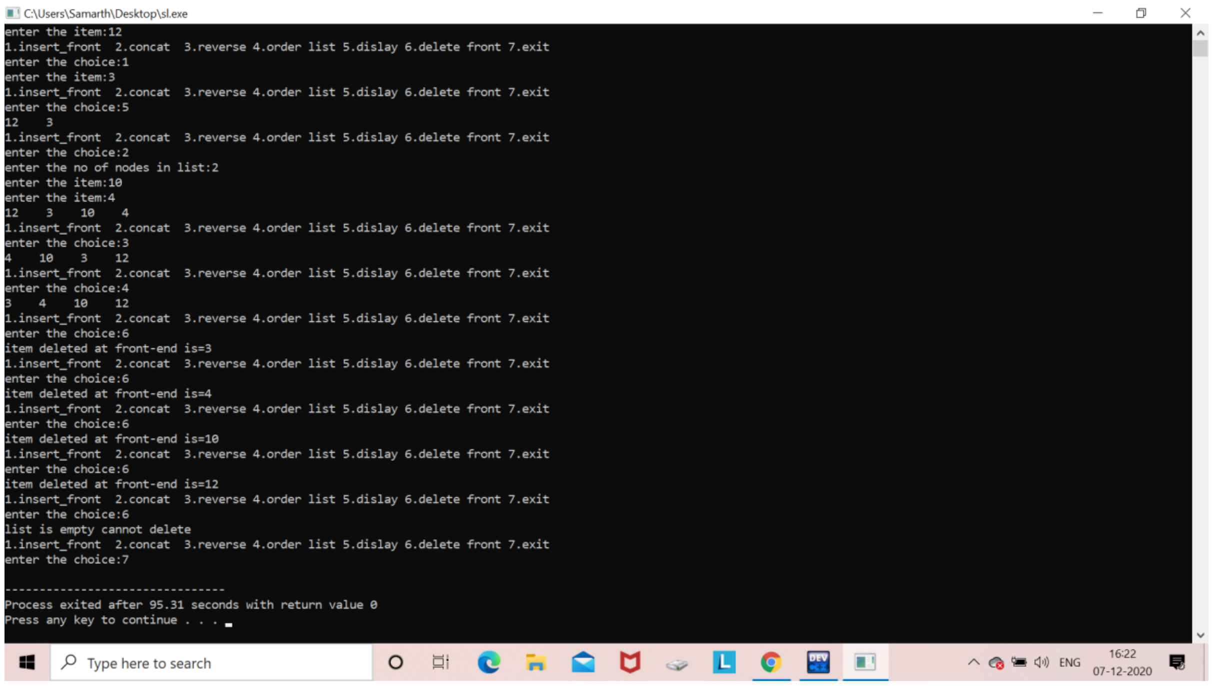
```c
//Program Singly linked list (concat,reverse,sort)
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<process.h>
struct node
 {
  int info;
  struct node *link;
 };
typedef struct node *NODE;
NODE getnode()
{
NODE x;
x=(NODE)malloc(sizeof(struct node));
if(x==NULL)
 {
  printf("mem full\n");
  exit(0);
 }
 return x;
}
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 \n");

 for(temp=first;temp!=NULL;temp=temp->link)
  {
  printf("%d    ",temp->info);
  }
  printf("\n");
```

```
}
NODE concat(NODE first,NODE second)
{
 NODE cur;
 if(first==NULL)
  return second;
 if(second==NULL)
  return first;
 cur=first;
 while(cur->link!=NULL)
  cur=cur->link;
 cur->link=second;
 return first;
}
NODE reverse(NODE first)
 {
 NODE cur,temp;
 cur=NULL;
 while(first!=NULL)
  {
   temp=first;
   first=first->link;
   temp->link=cur;
   cur=temp;
  }
 return cur;
}
    NODE sortList(NODE first) {
        NODE current = first, index = NULL;
        int temp;
        if(first == NULL) {
              printf("list is empty.");
            return current;
        }
        else {
            while(current != NULL) {

                index = current->link;

                while(index != NULL) {

                    if(current->info > index->info) {
                        temp = current->info;
                        current->info = index->info;
                        index->info = temp;
                    }
                    index = index->link;
                }
                current = current->link;
            }
                                    return current;
        }
    }

int main()
{
int item,choice,pos,i,n;
NODE first=NULL,a,b;
for(;;)
{
printf("1.insert_front  2.concat  3.reverse 4.order list 5.dislay 6.delete front 7.exit\n");
printf("enter the choice:");
scanf("%d",&choice);
switch(choice)
 {
  case 1:printf("enter the item:");
         scanf("%d",&item);
         first=insert_rear(first,item);
         break;
  case 2:printf("enter the no of nodes in list:");
         scanf("%d",&n);
         a=NULL;
         for(i=0;i<n;i++)
          {
           printf("enter the item:");
           scanf("%d",&item);
           a=insert_rear(a,item);
          }
          first=concat(first,a);
          display(first);
         break;
  case 3:first=reverse(first);
         display(first);
         break;
  case 4:sortList(first);
                          display(first);
         break;
  case 5:display(first);
         break;
  case 6:first=delete_front(first);
     break;
  default:exit(0);
  }

  }

  return 0;

  }
```



```
enter the item:12
1.insert_front  2.concat  3.reverse 4.order list 5.dislay 6.delete front 7.exit
enter the choice:1
enter the item:3
1.insert_front  2.concat  3.reverse 4.order list 5.dislay 6.delete front 7.exit
enter the choice:5
12    3
1.insert_front  2.concat  3.reverse 4.order list 5.dislay 6.delete front 7.exit
enter the choice:2
enter the no of nodes in list:2
enter the item:10
enter the item:4
12    3    10    4
1.insert_front  2.concat  3.reverse 4.order list 5.dislay 6.delete front 7.exit
enter the choice:3
4    10    3    12
1.insert_front  2.concat  3.reverse 4.order list 5.dislay 6.delete front 7.exit
enter the choice:4
3    4    10    12
1.insert_front  2.concat  3.reverse 4.order list 5.dislay 6.delete front 7.exit
enter the choice:6
item deleted at front-end is=3
1.insert_front  2.concat  3.reverse 4.order list 5.dislay 6.delete front 7.exit
enter the choice:6
item deleted at front-end is=4
1.insert_front  2.concat  3.reverse 4.order list 5.dislay 6.delete front 7.exit
enter the choice:6
item deleted at front-end is=10
1.insert_front  2.concat  3.reverse 4.order list 5.dislay 6.delete front 7.exit
enter the choice:6
item deleted at front-end is=12
1.insert_front  2.concat  3.reverse 4.order list 5.dislay 6.delete front 7.exit
enter the choice:6
list is empty cannot delete
1.insert_front  2.concat  3.reverse 4.order list 5.dislay 6.delete front 7.exit
enter the choice:7

--------------------------------
Process exited after 95.31 seconds with return value 0
Press any key to continue . . .
```