Document Project Build Tools Help

Save All | Revert | Close | Back | Forward | Compile | Build | Execute | Color Chooser | Find

stack.c | conversion.c | evaluation.c | factorial.c | gcd.c | queue.c | circular queue.c | priorityqueue.c | ascending.c | descending.c

```c
#include<stdio.h>
#include<conio.h>
#include<process.h>
#include<stdlib.h>
struct node{
    int inf;
    struct node *link;
};
typedef struct node *NODE;
NODE getnode()
{
    NODE x;
    x=(NODE)malloc(sizeof(struct node));
    if(x==NULL)
    {
        printf("MEMORY IS FULL\n");
        exit(0);
    }
    else
    return x;
}
NODE insertfront(NODE first,int item)
{
    NODE temp;
    temp=getnode();
    temp->inf=item;
    temp->link=NULL;
    if(first==NULL)
    {
        return temp;
    }
    temp->link=first;
    first=temp;
    return first;
}
NODE insertrear(NODE first,int item)
```

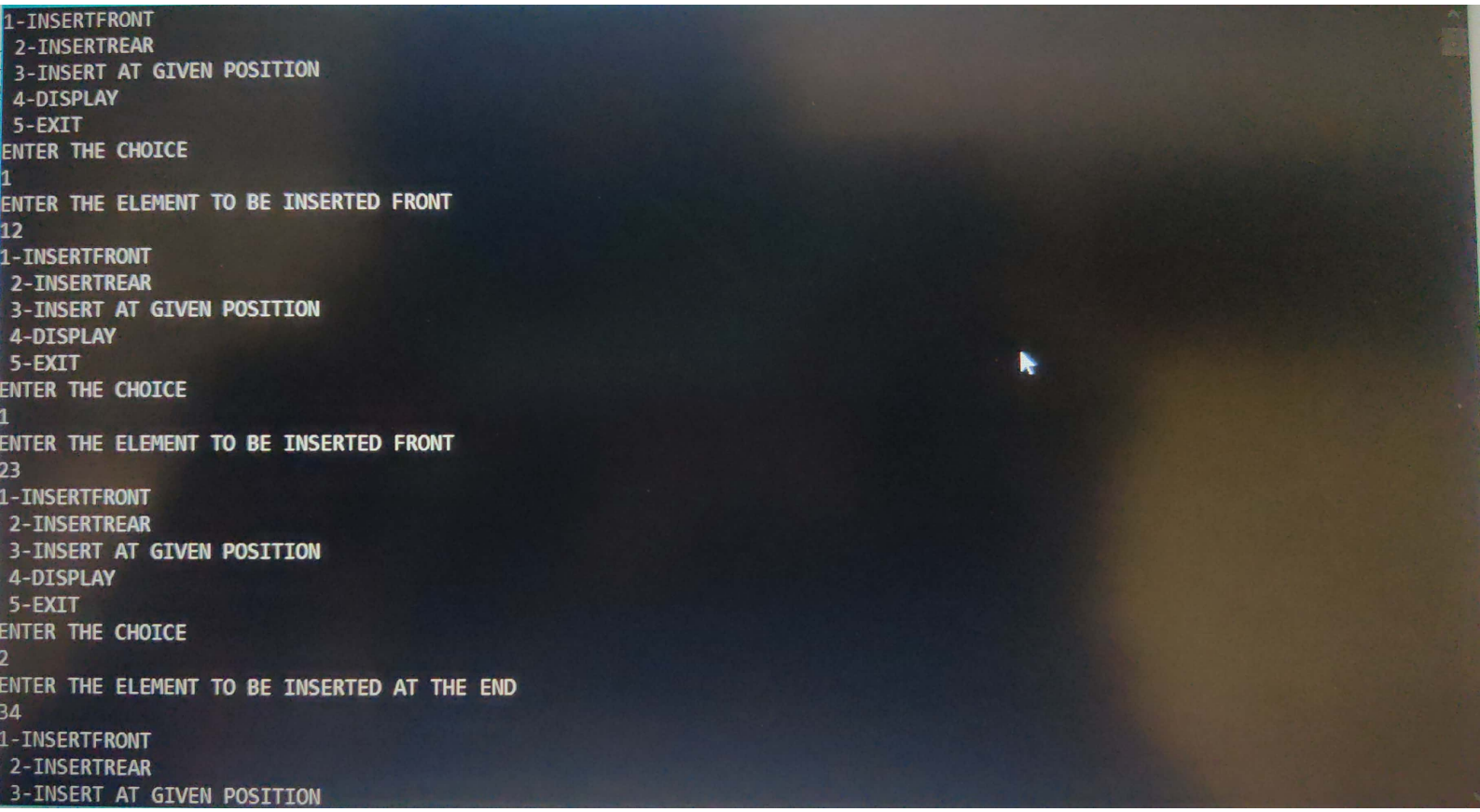
```
NODE insertrear(NODE first,int item)
{
    NODE temp,cur;
    temp=getnode();
    temp->inf=item;
    temp->link=NULL;
    if(first==NULL)
    {
        return temp;
    }
    cur=first;
    while(cur->link != NULL)
    {
        cur=cur->link;
    }
    cur->link=temp;
    return first;
}

NODE insertpos(int item,int pos,NODE first)
{
NODE temp,cur,prev;
int count;
temp=getnode();
temp->inf=item;
temp->link=NULL;
if(first==NULL&&pos==1)
{
return temp;
}
if(first==NULL)
{
printf("invalid position\n");
return first;
}
if(pos==1)
```

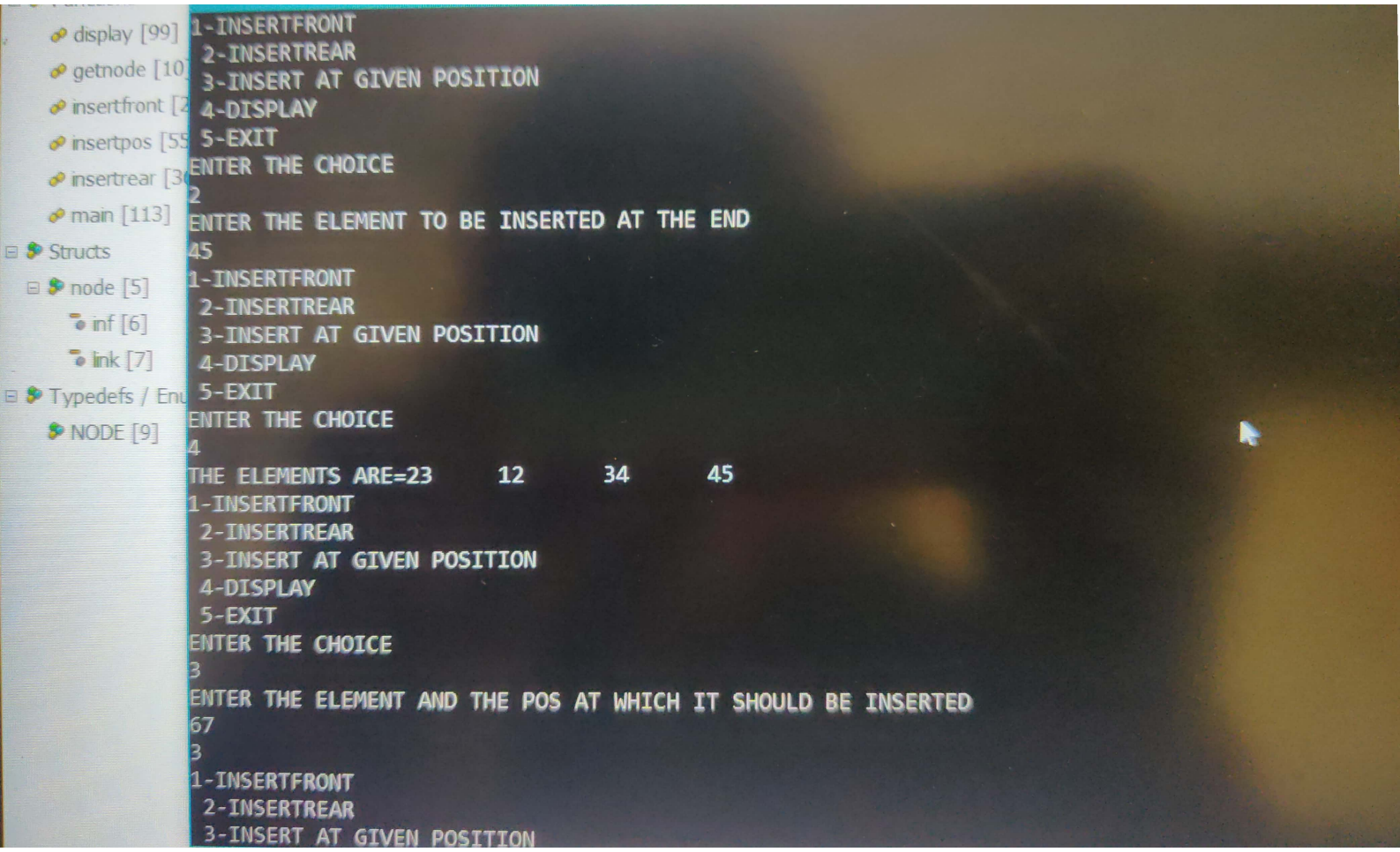
```
69      return first;
70      }
71      if(pos==1)
72      {
73      temp->link=first;
74      first=temp;
75      return temp;
76      }
77      count=1;
78      prev=NULL;
79      cur=first;
80      while(cur!=NULL&&count!=pos)
81      {
82      prev=cur;
83      cur=cur->link;
84      count++;
85      }
86      if(count==pos)
87      {
88
89      prev->link=temp;
90      temp->link=cur;
91      return first;
92      }
93      printf("invalid position\n");
94      return first;
95      }
96
97
98
99      void display(NODE first)
100     {
101         NODE temp;
102         if(first==NULL){
103             printf("THE LIST IS EMPTY\n");
104         }
```

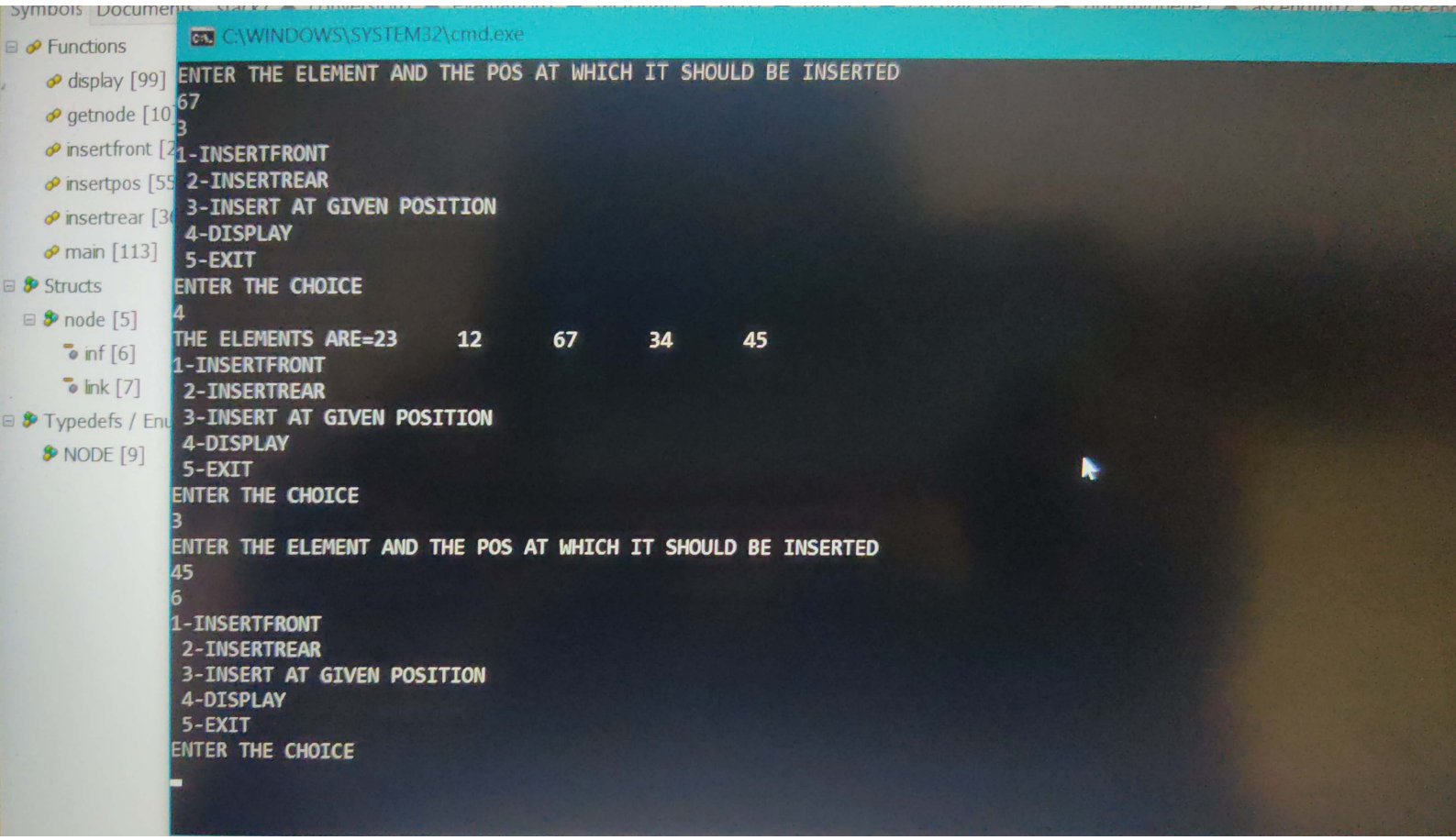
```
ack.c ✖ conversion.c ✖ evaluation.c ✖ factorial.c ✖ gcd.c ✖ queue.c ✖
106     for(temp=first;temp!=NULL;temp=temp->link)
107     {
108         printf("%d\t",temp->inf);
109     }
110     printf("\n");
111 }
112
113 int main()
114 {
115     int c,item,pos;
116     NODE first=NULL;
117     for(;;)
118     {
119         printf("1-INSERTFRONT \n 2-INSERTREAR \n 3-INSERT AT GIVEN POSITION \n 4-DISPLAY \n 5-E
120         printf("ENTER THE CHOICE\n");
121         scanf("%d",&c);
122         switch(c)
123         {
124             case 1:
125             printf("ENTER THE ELEMENT TO BE INSERTED FRONT\n");
126             scanf("%d",&item);
127             first=insertfront(first,item);
128             break;
129             case 2:
130             printf("ENTER THE ELEMENT TO BE INSERTED AT THE END\n");
131             scanf("%d",&item);
132             first=insertrear(first,item);
133             break;
134               case 3:
135               printf("ENTER THE ELEMENT AND THE POS AT WHICH IT SHOULD BE INSERTED\n");
136             scanf("%d",&item);
137             scanf("%d",&pos);
138             first=insertpos(item,pos,first);
139             break;
140             case 4:
141             display(first);
```

```
            printf("ENTER THE ELEMENT TO BE INSERTED FRONT\n");
            scanf("%d",&item);
            first=insertfront(first,item);
            break;
            case 2:
            printf("ENTER THE ELEMENT TO BE INSERTED AT THE END\n");
            scanf("%d",&item);
            first=insertrear(first,item);
            break;
              case 3:
              printf("ENTER THE ELEMENT AND THE POS AT WHICH IT SHOULD BE INSERTED\n");
            scanf("%d",&item);
            scanf("%d",&pos);
            first=insertpos(item,pos,first);
            break;
            case 4:
            display(first);
            break;
            case 5:
            exit(0);
          default:
          printf("INVALID CHOICE\n");
      }
  }
}
```

```
kedlist1" "linkedlist1.c" (in directory: C:\Users\Nithin')
hed successfully.
```


1-INSERTFRONT
2-INSERTREAR
3-INSERT AT GIVEN POSITION
4-DISPLAY
5-EXIT
ENTER THE CHOICE
1
ENTER THE ELEMENT TO BE INSERTED FRONT
12
1-INSERTFRONT
2-INSERTREAR
3-INSERT AT GIVEN POSITION
4-DISPLAY
5-EXIT
ENTER THE CHOICE
1
ENTER THE ELEMENT TO BE INSERTED FRONT
23
1-INSERTFRONT
2-INSERTREAR
3-INSERT AT GIVEN POSITION
4-DISPLAY
5-EXIT
ENTER THE CHOICE
2
ENTER THE ELEMENT TO BE INSERTED AT THE END
34
1-INSERTFRONT
2-INSERTREAR
3-INSERT AT GIVEN POSITION


display [99]
getnode [10
insertfront [2
insertpos [55
insertrear [3
main [113]
Structs
node [5]
inf [6]
link [7]
Typedefs / Enu
NODE [9]
1-INSERTFRONT
2-INSERTREAR
3-INSERT AT GIVEN POSITION
4-DISPLAY
5-EXIT
ENTER THE CHOICE
2
ENTER THE ELEMENT TO BE INSERTED AT THE END
45
1-INSERTFRONT
2-INSERTREAR
3-INSERT AT GIVEN POSITION
4-DISPLAY
5-EXIT
ENTER THE CHOICE
4
THE ELEMENTS ARE=23 12 34 45
1-INSERTFRONT
2-INSERTREAR
3-INSERT AT GIVEN POSITION
4-DISPLAY
5-EXIT
ENTER THE CHOICE
3
ENTER THE ELEMENT AND THE POS AT WHICH IT SHOULD BE INSERTED
67
3
1-INSERTFRONT
2-INSERTREAR
3-INSERT AT GIVEN POSITION


Symbols
Functions
display [99]
getnode [10]
insertfront [2
insertpos [55
insertrear [3
main [113]
Structs
node [5]
inf [6]
link [7]
Typedefs / Enu
NODE [9]
C:\WINDOWS\SYSTEM32\cmd.exe
ENTER THE ELEMENT AND THE POS AT WHICH IT SHOULD BE INSERTED
67
3
1-INSERTFRONT
2-INSERTREAR
3-INSERT AT GIVEN POSITION
4-DISPLAY
5-EXIT
ENTER THE CHOICE
4
THE ELEMENTS ARE=23 12 67 34 45
1-INSERTFRONT
2-INSERTREAR
3-INSERT AT GIVEN POSITION
4-DISPLAY
5-EXIT
ENTER THE CHOICE
3
ENTER THE ELEMENT AND THE POS AT WHICH IT SHOULD BE INSERTED
45
6
1-INSERTFRONT
2-INSERTREAR
3-INSERT AT GIVEN POSITION
4-DISPLAY
5-EXIT
ENTER THE CHOICE