1985

جامعة محمد بوضياف - المسيلة
Université Mohamed Boudiaf - M'sila

LICENSE 2 IN COMPUTER SCIENCE 2022/2023

Algorithmique et Structures de Données 3

SERIES of EXERCISES 2

Solutions

10101010101010101010101010101010101010101010101010101010101010101010101010101010101010101010

EXERCISE 1: (Pointers)

```cpp
/////// Pointers in c++
#include<iostream>
using namespace std;
main()
{
    int *p=new int(61);
    cout<<*p<<endl;
    delete p;
    p=new int[5];
        for (int i=0;i<5;i++)cin>>p[i];
        for (int i=0;i<5;i++)cout<<p[i];
    delete []p;
}
```

```
"C:\Users\Raouf\Desktop\New folder\exo1 b.exe"
61
1
2
3
4
5
12345
Process returned 0 (0x0)   execution time : 7.416 s
Press any key to continue.
```

```cpp
/////// Pointers in c++
#include<iostream>
using namespace std;
main()
{
int a=4,t[10]={2,5,1,4,3,6,7,9,8,11};
int *p1=t,*p2= &a;
cout <<*p1<<endl;
cout <<*p1+*p2<<endl;
p2=p1; p1+= 5;
cout <<*p1+10<<endl;
cout <<p1-p2<<endl;
}
```

```
"C:\Users\Raouf\Desktop\New folder\exo1 a.exe"
2
6
16
5

Process returned 0 (0x0)   execution time : 0.036 s
Press any key to continue.
```

EXERCISE 2: (Linked lists)

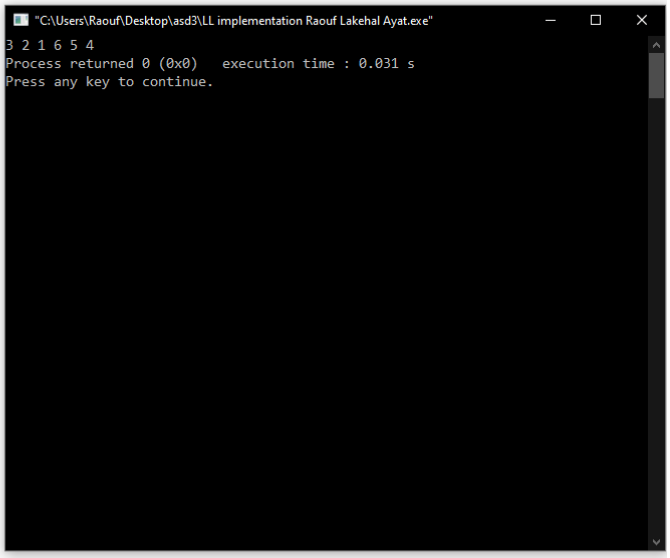
```cpp
struct node
{
int data;
node *next;
node *previous;
};

/* الدالة الخاصة بإنشاء رأس القائمة*/
node* head_creation()
{
   node *head=new node; // إنشاء وحجز مؤشر رأس القائمة
   head->next=NULL;     // المؤشر الموالي للرأس يأخذ نيل
   head->previous=NULL; // المؤشر السابق للرأس يأخذ نيل
   cout<<"enter a data value of the head\n";
   cin>>head->data;
   return head;         //  إرجاع قيمة رأس القائمة
}
```

Edited by: Raouf O. Lakehal Ayat **Course coordinator: hemza.loucif@univ-msila.dz**

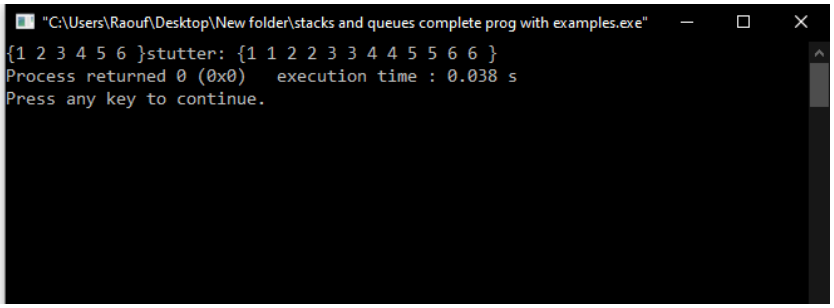
EXERCISE 3: (Linked lists)

```
1  #include<iostream>
2  struct node {     int   data;
3                    node *next;
4                 };
5  node* head_creation(int x)
6     {  node *head=new node;
7        head->next=NULL;
8        head->data=x;
9        return head;
10    }
11 node *add_nodes_head(int x,node *head)
12   {  node *p;
13      p=new node;
14      p->data=x;
15      p->next=head;
16      head=p;
17      return head;
18   }
19 main()
20 {  node *head1,*head2,*browser;
21    head1=head_creation(1);
22    head1=add_nodes_head(2,head1);
23    head1=add_nodes_head(3,head1);
24    head2=head_creation(4);
25    head2=add_nodes_head(5,head2);
26    head2=add_nodes_head(6,head2);
27    browser=head1;
28    while(browser->next!=NULL)
29    {browser=browser->next;}
30     browser->next=head2;  // concatenation
31     browser=head1;
32    while(browser!=NULL)
33    {std::cout<<browser->data<<" ";browser=browser->next;}
34 }
```



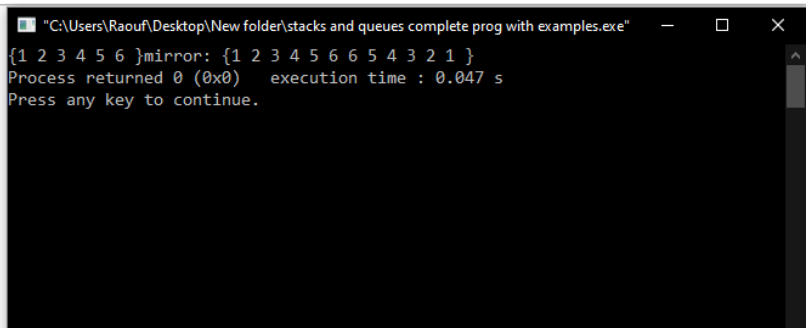
EXERCISE 4: (Stacks & Queues)

```
154  void STUTTER()
155  { int a,n=SIZE();
156    for(int i=1;i<=n;i++)
157    {
158     a=DEQUEUE();
159     ENQUEUE(a);
160     ENQUEUE(a);
161    }
162  }
163
```



EXERCISE 5: (Stacks & Queues)

```
165  void MIRROR()
166  { int n=SIZE();
167    int a[n];
168    for(int i=1;i<=n;i++)
169      a[i]=DEQUEUE();
170    for(int i=1;i<=n;i++)
171      ENQUEUE(a[i]);
172    for(int i=n;i>=1;i--)
173      ENQUEUE(a[i]);
174  }
175
```



EXERCISE 6: (Functions & recursion)

```cpp
#include<iostream>
using namespace std;
int* increment(int *t,int carry,int n)
{
  if (t[n-1]-carry!=0)
     t[n-1]=1;
   else
     {t[n-1]=0;
     increment(t,carry,n-1);
     }
   return t;
}
main()
{
int t[]={1,1,1,1,0,1,1,1};
cout<<"before\n";
for(int i=0;i<8;i++)    cout<<t[i];
increment(t,1,8);
cout<<"\nAfter\n";
for(int i=0;i<8;i++)    cout <<t[i];
}
```