EED 1010 ALGORITHMS & PROGRAMMING

Name: Enes

Surname: Erten

Number: 2018502036

Date: 24.04.2020

Lab Task 9

Lab Section: 4(LAB), 1(Theory)

**Task 1**: Write a program that inserts 20 random integers from 0 to 100 in order in a linked list. The program should calculate the sum of the elements and the floating point average of the elements. **Note: Do not use any sorting algorithms.**.

```
The list with 20 random integers:
  5   9 11 12 13 16 17 25 37 38 45 49 53 58 63 74 80 86 88 91

sum of the integers: 870

average of the integers: 43.50
```

Figure 1.1: Output of Laboratory study task1

The Code:

```
#include<stdio.h>

#include<stdlib.h>

#include<time.h>


typedef struct listnode{

   int num;

   struct listnode *next;

}listnode;

//struct defintion

//firs elemnt is keeping an element

//second one is pointing new listnode

//chanhge name by typedef


typedef listnode *listnodeptr;

//change name by typedef
```

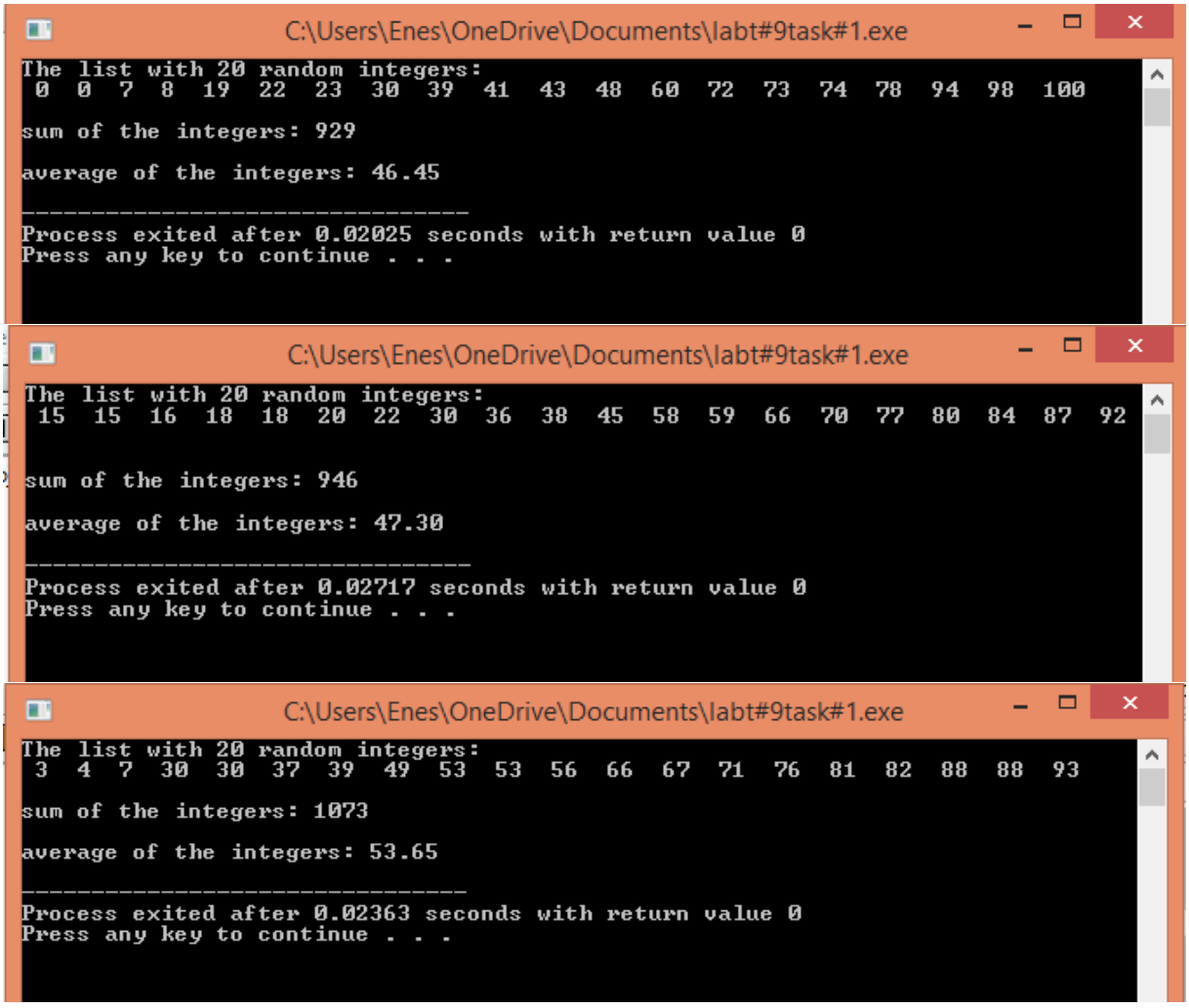
```
void insertlist(listnodeptr *a, int n);
void printlist(listnodeptr a);
int sumofl(listnodeptr a);
double avrgofl(int n,int n1);
//function prototype

//this functions inserts the elements to the list
void insertlist(listnodeptr *a,int n)
{
   listnodeptr nptr,pptr,cptr;
   //dynamic memory allocation
   nptr=(listnode *)malloc(sizeof(listnode));

   if(nptr!=NULL)
   {
      nptr->num=n;
      nptr->next=NULL;

      pptr=NULL;
      cptr=*a;

      while(cptr!=NULL && n>cptr->num)
          {
         pptr=cptr;
         cptr=cptr->next;
      }
       //finding the number locataions to insert

      if(pptr==NULL)
          {
         nptr->next=*a;
```

```
        *a=nptr;
      }
      //assign number if the number is the smallest elemnt int list
      else
            {//else assign nu,mber to the list
        pptr->next=nptr;
        nptr->next=cptr;
      }
   }

         else
            printf("No memory available!\n");
}

//this functions prints the list
void printlist(listnodeptr a)
{
   if(a==NULL)
      printf("List is empty.\n");

   else
         {
      while (a!=NULL)
                  {
        printf(" %d ",a->num);
        a=a->next;
      }
      //printing elemnts one by one
   }
}
```

```
//this functions returns the total of elements
//on the lists
int sumofl(listnodeptr a)
{
   listnodeptr a1;
   int sum=0;

   a1=a;

   while(a1!=NULL)
          {
      sum+=a1->num;
      a1=a1->next;
   } //sums elements of list

   return sum;
}

double avrgofl(int n,int n1)
{
          return (float)n/n1;
}

int main()
{
   listnodeptr list=NULL;
   int i;

   srand(time(NULL));
    //to generate random number
   for (i=0;i<20;i++)
```

```
    insertlist(&list,rand()%101);
  //send to the function insert elemnts of list
  printf("The list with 20 random integers:\n");
  printlist(list);
  //send to the function to print list
  printf("\n\nsum of the integers: %d\n\n",sumofl(list));
  printf("average of the integers: %.2f\n\n",avrgofl(sumofl(list),20));
  //print the avrg of list and sum of list
}
```

The Outputs:



**Task 2**: Write a program that deletes a number from the list (in Task1) and frees the memory associated with this node. The user firstly must print the list of 20 random integer numbers and then enters a number which he/she wants to delete. After deleting part, the linked list must be printed again.

The code:

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>

typedef struct listnode{
   int num;
   struct listnode *next;
}listnode;
//struct defintion
//firs elemnt is keeping an element
//second one is pointing new listnode
//chanhge name by typedef

typedef listnode *listnodeptr;
//change name by typedef

void insertlist(listnodeptr *a, int n);
void printlist(listnodeptr a);
int isempty(listnodeptr a);
int deletelist(listnodeptr *a,int n);
int sumofl(listnodeptr a);
double avrgofl(int n,int n1);
//function prototype

//this functions inserts the elements to the list
void insertlist(listnodeptr *a,int n)
```

```
{
   listnodeptr nptr,pptr,cptr;
   //dynamic memory allocation
   nptr=(listnode *)malloc(sizeof(listnode));

   if(nptr!=NULL)
   {
      nptr->num=n;
      nptr->next=NULL;

      pptr=NULL;
      cptr=*a;

      while(cptr!=NULL && n>cptr->num)
           {
         pptr=cptr;
         cptr=cptr->next;
      }
       //finding the number locataions to insert

      if(pptr==NULL)
           {
         nptr->next=*a;
         *a=nptr;
      }
      //assign number if the number is the smallest elemnt int list
      else
            {//else assign nu,mber to the list
         pptr->next=nptr;
         nptr->next=cptr;
      }
```

```
    }

            else
                printf("No memory available!\n");
}

//this functions prints the list
void printlist(listnodeptr a)
{
    if(a==NULL)
        printf("List is empty.\n");

    else
            {
        while (a!=NULL)
                        {
            printf(" %d ",a->num);
            a=a->next;
        }
        //printing elemnts one by one
    }
}

//if list is empty return 0 else returns 0
int isempty(listnodeptr a)
{
            if(a!=NULL)
                return 1;

            else
                return 0;
```

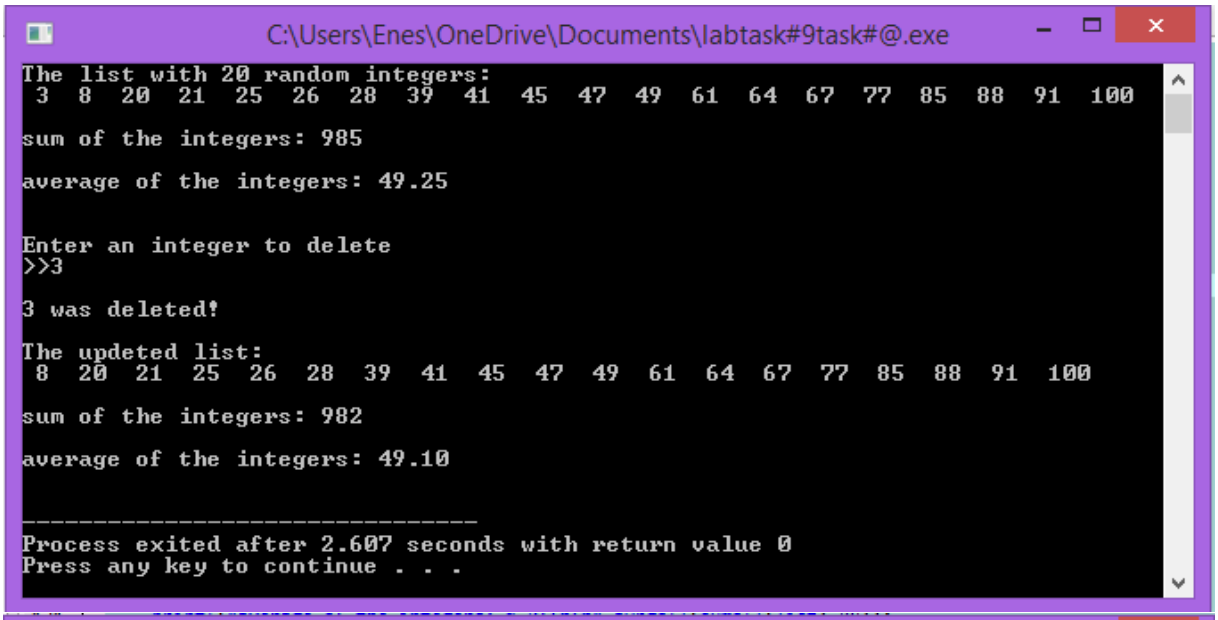
```
}

//this function is delete an elemnt from list
//if elemnt found in list returns 1 else returns 0
int deletelist(listnodeptr *a,int n)
{
        listnodeptr pptr,cptr,tmptr;

        if(n==(*a)->num)
        {//if element is equal to first element
                tmptr=*a;
                *a=(*a)->next;
                free(tmptr);

                return 1;
   }

        else
        {//finding the elemnt
                pptr=*a;
                cptr=(*a)->next;

                while(cptr!=NULL && cptr->num!=n)
                {
                        pptr=cptr;
                        cptr=cptr->next;
                }


      if(cptr!=NULL)
      {
```

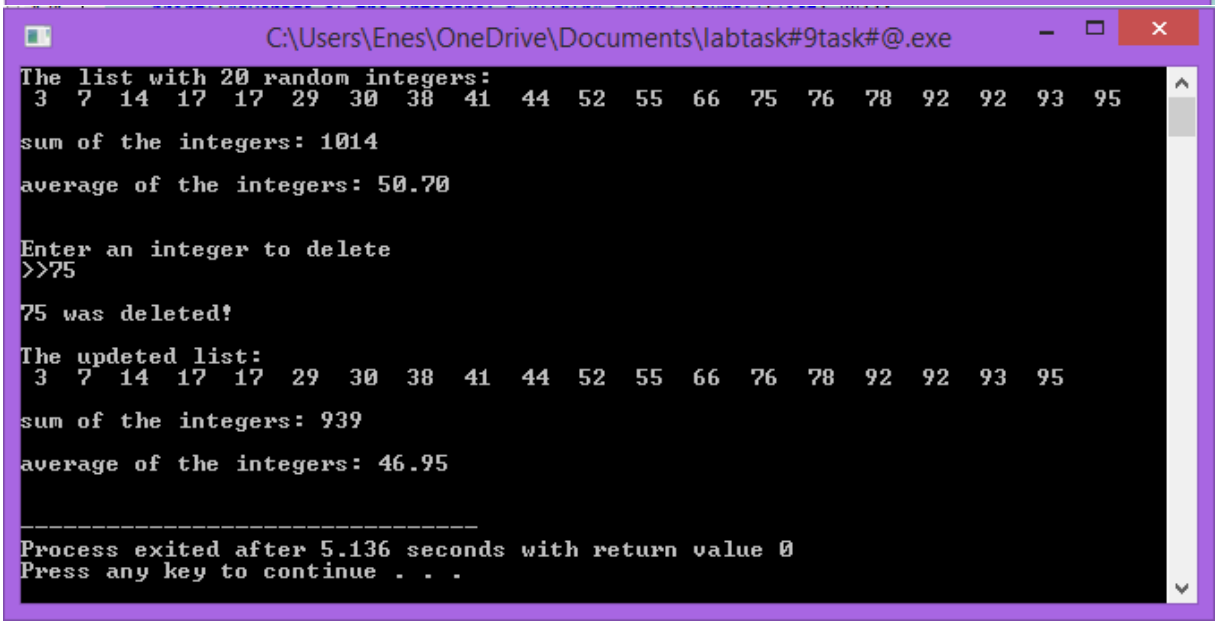
```
            tmptr=cptr;
            pptr->next=cptr->next;
            free(tmptr);

            return 1;
            }//if element found in the list

            else //else
               return 0;
   }


}

//this functions returns the total of elements
//on the lists
int sumofl(listnodeptr a)
{
   listnodeptr a1;
   int sum=0;

   a1=a;

   while(a1!=NULL)
         {
      sum+=a1->num;
      a1=a1->next;
   } //sums elements of list

   return sum;
}
```

```
double avrgofl(int n,int n1)
{
        return (float)n/n1;
}

int main()
{
   listnodeptr list=NULL;
   int i,z;

   srand(time(NULL));
    //to generate random number
   for (i=0;i<20;i++)
      insertlist(&list,rand()%101);
   //send to the function insert elemnts of list
   printf("The list with 20 random integers:\n");
   printlist(list);
   //send to the function to print list
   printf("\n\nsum of the integers: %d\n\n",sumofl(list));
   printf("average of the integers: %.2f\n\n",avrgofl(sumofl(list),20));
   //print the avrg of list and sum of list

      if(isempty(list))
   {//if list is not empty
        printf("\nEnter an integer to delete\n>>");
        scanf("%d",&z);
        printf("\n");
        if(deletelist(&list,z))
        {//if function returns 1
              printf("%d was deleted!\n\n",z);
```

```
                printf("The updeted list:\n");
                        printlist(list);//print list
                   printf("\n\nsum of the integers: %d\n\n",sumofl(list));
          printf("average of the integers: %.2f\n\n",avrgofl(sumofl(list),20));
          //print the avrg of updated list and sum of updaeted list
                }

                else
                   printf("the number doesnt exist in the list!\n\n");
                   //if function returns zero
           }

           else//if list is empty
              printf("list is empty!\n\n");
}
```

The Outputs:


C:\Users\Enes\OneDrive\Documents\labtask#9task#@.exe
The list with 20 random integers:
3 8 20 21 25 26 28 39 41 45 47 49 61 64 67 77 85 88 91 100
sum of the integers: 985
average of the integers: 49.25
Enter an integer to delete
>>3
3 was deleted!
The updeted list:
8 20 21 25 26 28 39 41 45 47 49 61 64 67 77 85 88 91 100
sum of the integers: 982
average of the integers: 49.10
Process exited after 2.607 seconds with return value 0
Press any key to continue . . .


C:\Users\Enes\OneDrive\Documents\labtask#9task#@.exe
The list with 20 random integers:
3 7 14 17 17 29 30 38 41 44 52 55 66 75 76 78 92 92 93 95
sum of the integers: 1014
average of the integers: 50.70
Enter an integer to delete
>>75
75 was deleted!
The updeted list:
3 7 14 17 17 29 30 38 41 44 52 55 66 76 78 92 92 93 95
sum of the integers: 939
average of the integers: 46.95
Process exited after 5.136 seconds with return value 0
Press any key to continue . . .