Національний технічний університет України
"Київський політехнічний інститут імені Ігоря Сікорського"
Факультет Електроніки
Кафедра мікроелектроніки

ЗВІТ
Про виконання лабораторної роботи №3
з дисципліни: «Алгоритми та структури даних-2»

Робота зі списками. Динамічна зміна розмірів списків

Виконавець:
Студент 3-го курсу _____ А. С. Мнацаканов
(підпис)

Перевірив: _____ Д. Д. Татарчук
(підпис)

2020

Мета роботи – закріпити навички використання списків при розробці програм, вивчити прийоми динамічного розподілу пам'яті.

Завдання

Написати програму, що виконує наступні дії:

1) Зчитує дані із файлу отриманого в першій лабораторній роботі та зберігає їх у пам'яті у вигляді списку заданого у таблиці 4 відповідно до варіанту завдань.

2) Виводить дані на екран у вигляді двох стовпців x f(x) , розділених символами табуляції. Стовпці повинні мати заголовки X та Y відповідно.

3) Обчислює значення функції у точках, що знаходяться посередині між сусідніми точками отриманими з файлу, як середнє значення двох сусідніх значень і добавляє ці точки до свого списку.

4) Виводить отриманий список на екран, як визначено у п.2.

5) Видаляє із списку 5 елементів, що містять дані введені з клавіатури та відображає отриманий список на екрані згідно п. 2.

Виконання роботи

Код на C++

```
#include <fstream>
#include <iostream>
#include <cmath>

class spisok_elem{
public:
spisok_elem(double x, double y):
  x(x),
  y(y),
  next(NULL),
  prev(NULL){}
  ~spisok_elem(){
    if(next){
      delete next;
    }
  }

  void set_x(double x1){
            x = x1;
  }
  void set_y(double y1){
```

```
        y = y1;
}
void set(double x1, double y1){
        x = x1;
        y = y1;
}

double get_x(){
        return x;
}
double get_y(){
        return y;
}

void set_next(spisok_elem* elem){
        next = elem;
}

spisok_elem* get_next(){
        return next;
}

void set_prev(spisok_elem* elem){
        prev = elem;
}

spisok_elem* get_prev(){
        return prev;
}


spisok_elem* search(double xx){
  if(xx>(x-0.001) && xx<(x+0.001))
  {
    return this;
  }
  if(!next){
    return NULL;
  }
  return next->search(xx);
}

void add_after(spisok_elem* node){
  if(next == NULL){
    next = node;
    node-> prev = this;
    node -> next = NULL;
  }
  else
  {
    spisok_elem* temp = next;
    next = node;
    node-> prev = this;
```

```
      node -> next = temp;
      temp -> prev = node;
    }
  }

  spisok_elem* remove(){
    if(next){
      next-> prev = prev;
    }
    if(prev){
      prev-> next = next;
    }
  }

  int size(int k = 0){
    if(next == NULL)
    {return k+1;}
    return next->size(k+1);
  }

void p3(){
  if(next){
    next->p3();
    spisok_elem* sp = new spisok_elem(x+(next->x-x)/2,  sin(fabs(x
       +(next->x-x)/2)));
    add_after(sp);
  }
}


  private:
  double x;
  double y;
  spisok_elem* next;
  spisok_elem* prev;
};

class spisok{
public:
  spisok ():
    begin(NULL),
    end(NULL){}

~spisok(){
  if(begin){
    delete begin;
  }
}

  spisok* add(spisok_elem* to_add, spisok_elem* before = NULL){
    if( before == NULL ){
      return push_back(to_add);
    }
```

```
    before->add_after(to_add);
    return this;
  }

  spisok* push_back(spisok_elem* to_add){
    if(begin == NULL && end == NULL){
      begin = end = to_add;
      begin -> set_next(NULL);
      end -> set_prev(NULL);
      return this;
    }
    end-> set_next(to_add);
    to_add-> set_prev(end);
    to_add-> set_next(NULL);
    end = to_add;
    return this;
  }

  spisok* push_back(double x, double y)
{
  spisok_elem* new_sp = new spisok_elem(x,y);
  return push_back(new_sp);
}
spisok_elem* find(double x){
if(begin){
  return begin-> search(x);
}
return NULL;
}

spisok_elem* remove(double x){
  spisok_elem* to_remove = find(x);
  if(! to_remove)
  {
    return NULL;
  }
  if(begin== to_remove){
    begin= to_remove-> get_next();
  }
  if(end == to_remove)
  {
    end = to_remove-> get_prev();
  }
  to_remove-> remove();
  return to_remove;
}

spisok_elem* get_begin(){
  return begin;
}

size_t size(){
  if(begin)
```

```
    return begin -> size();
  return 0;
}

spisok* p3(){
  if(begin)
    begin->p3();
  return this;
}

private:
  spisok_elem* begin;
  spisok_elem* end;
};

std::ostream& operator << (std::ostream& s, spisok_elem& l){
    s << l.get_x() << " \t " << l.get_y() << std::endl;
    if(l.get_next() ){
      s << *(l.get_next());
    }
    return s;
}

std::ostream& operator << (std::ostream& s, spisok& l){
    s << "X \t\t Y" << std::endl;
    if(l.get_begin() ){
      s << *(l.get_begin());
    }
    return s;
}

int main(){
  spisok list;
  std::ifstream in("DP8205.txt");
  while(!in.eof()){
    double r_x, r_y;
          in >> r_x >> r_y;
    list.push_back(r_x, r_y);
  }
  std::cout << list << std::endl;

  list.p3();

  std::cout << list << std::endl;

  for(int i=0; i<5;++i){
    double ii;
    std::cout << "input data to delete: ";
    std::cin >> ii;
    if(list.remove(ii)){
      std::cout << "removed" << std::endl;
    }else{
      std::cout << "not found" << std::endl;
```

```
    }
  }

  std::cout << list << std::endl;
  return 0;
}
```

```
X               Y
0       0
0.0785398       0.0784591
0.15708         0.156434
0.235619        0.233445
0.314159        0.309017
0.392699        0.382683
0.471239        0.45399
0.549779        0.522499
0.628319        0.587785
0.706858        0.649448
0.785398        0.707107
0.785398        0.707107

X               Y
0       0
0.0392699       0.0392598
0.0785398       0.0784591
0.11781         0.117538
0.15708         0.156434
0.19635         0.19509
0.235619        0.233445
0.274889        0.27144
0.314159        0.309017
0.353429        0.346117
0.392699        0.382683
0.431969        0.41866
0.471239        0.45399
0.510509        0.488621
0.549779        0.522499
0.589049        0.555571
0.628319        0.587785
0.667588        0.619094
0.706858        0.649448
0.746128        0.678801
0.785398        0.707107
0.785398        0.707107
0.785398        0.707107

input data to delete:
```

до

```
input data to delete: 0
removed
input data to delete: 0.0392699
removed
input data to delete: 0.11781
removed
input data to delete: 0.235619
removed
input data to delete: 0.353429
removed
X               Y
0.0785398       0.0784591
0.15708         0.156434
0.19635         0.19509
0.274889        0.27144
0.314159        0.309017
0.392699        0.382683
0.431969        0.41866
0.471239        0.45399
0.510509        0.488621
0.549779        0.522499
0.589049        0.555571
0.628319        0.587785
0.667588        0.619094
0.706858        0.649448
0.746128        0.678801
0.785398        0.707107
0.785398        0.707107
0.785398        0.707107
```

та після видалення