```
/*****************************************************************
   ***************
*
    *
*   Porządkowanie wierzchołków wg rosnących kątów nachylenia ich
    *
*   wektorów wodzących                              *
*   Plik pobrany ze strony:           www.algorytm.org
    *
*   Opracowal:              Michal Knasiecki      *
*
    *
*****************************************************************
   ***************/

/*Uwaga:

W poniższej implementacji posłużono się biblioteką STL, która
zawiera szablon wektora.
Jeżeli twój kompilator nie posiada tej biblioteki możesz zastąpić
wektor tablicą.

*/



#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

#include <vector.h>
#include <algorithm>

//Struktura reprezentująca punkt
typedef struct {
           int x;
        int y;
        } Tpunkt;

//Struktura wyniku: punkt oraz współczynnik alfa
typedef struct {
    int x;
        int y;
        float alfa;
         } Twynik;


using namespace std;
```

```
//Wektor wynikowy
vector <Twynik> wynik;

//Funkcja sortująca wektor wynik wg współczynnika alfa
void quicksort_1(int x, int y)
{
int i,j;
float v;
Twynik temp;
i=x;
j=y;
v=wynik[div(x+y,2).quot].alfa;
do
   {
   while (wynik[i].alfa<v) i++;
   while (v<wynik[j].alfa) j--;
   if (i<=j)
    {
      temp=wynik[i];
      wynik[i]=wynik[j];
      wynik[j]=temp;
      i++;
      j--;
      }
   }
while (i<=j);
if (x<j) quicksort_1(x,j);
if (i<y) quicksort_1(i,y);
}

//Funkcja sortująca wektor wynik wg współrzędnej x (punkty o takim
   samym alfa)
void quicksort_2(int x, int y)
{
int i,j;
float v;
Twynik temp;
i=x;
j=y;
v=wynik[div(x+y,2).quot].x;
do
   {
   while (wynik[i].x<v) i++;
   while (v<wynik[j].x) j--;
   if (i<=j)
    {
      temp=wynik[i];
```

```
      wynik[i]=wynik[j];
      wynik[j]=temp;
      i++;
      j--;
      }
   }
while (i<=j);
if (x<j) quicksort_2(x,j);
if (i<y) quicksort_2(i,y);
}


/*Funkcja szuka w wektorze przedziału <i,j>, w którym punkty
  mają taki sam współczynnik alfa, a na stepnie sortuje ten
  przedział wg współrzędnej x*/
void sortuj(void)
{
int i=0;
int j;

//Znajdź przedział <i,j> o wspólnym alfa
do
 {
 if (wynik[i].alfa==wynik[i+1].alfa)
    {
   j=i;
   do
    j++;
   while (wynik[i].alfa==wynik[j].alfa);
   j--;

//Posortuj ten przedział
   quicksort_2(i,j);
   i=j;
   } else
   i++;
 }
 while (i<wynik.size()-1);
}



//-----------------------------------------

void main(void){

//Wektor wejściowy
vector <Tpunkt> wektor;
```

```
//Zmienne pomocnicze
Tpunkt p;
Twynik w;

//Czynnik d=|x|+|y|
float d;

//Współczynnik alfa
float alfa;


//Wprowadzamy wartości wejściowe do wektora
//Przykladowe wartosci (takie, jak na stronie)
p.x=-5;
p.y=-1;
wektor.push_back(p);
p.x=4;
p.y=-4;
wektor.push_back(p);
p.x=2;
p.y=4;
wektor.push_back(p);
p.x=-3;
p.y=2;
wektor.push_back(p);
p.x=4;
p.y=1;
wektor.push_back(p);

//Obliczamy wartosc d oraz alfa
while (!(wektor.empty())){

    p=wektor.back();
    wektor.pop_back();
    d=abs(p.x)+abs(p.y);

    if ((p.x>=0)&&(p.y>=0))
     alfa=p.y/d;

    if ((p.x<0)&&(p.y>=0))
     alfa=2-p.y/d;

    if ((p.x<0)&&(p.y<0))
     alfa=2+abs(p.y)/d;

    if ((p.x>=0)&&(p.y<0))
     alfa=4-abs(p.y)/d;
```

```
    //Zapisz punkt i wspolczynnik do wektora wynikowego
    w.x=p.x;
    w.y=p.y;
    w.alfa=alfa;
    wynik.push_back(w);
}

//Posortuj wektor wynikowy wg wartości czynnika alfa
quicksort_1(0,wynik.size()-1);

//Posortuj punkty o takim samym alfa wg współrzędnej x
sortuj();

//Wypisz wynik
for (int i=0;i<wynik.size();i++)
{
w=wynik[i];
printf("Punkt #%d: (%-2d,%-2d)\talfa=%.3f\n",i+1,w.x,w.y,w.alfa);
}

getch();
return;
}
```