



תכנות פורטרוז ו C++ פתרון מבתן Complex

מבחן בקורס תכנות C++ לתלמידי פיזיקה - 76641

מועד א' סמסטר ב' תש"ס

כל חומר מותר

משך הזמן שעתיים

ביח"ר לאריחים מייצר סוגים שונים של אריחי רצפה לריצוף. מבנה הנתונים השמור לכל אריח הינו מחלקה של ++C בעלת המבנה הבא:

```
01 class balata
02 {
03     char *name;
04     int width;
05     int length;
06     double price;
07 };
```

לכל אריח יש שם (name).

האריחים הם מלבניים בעלי אורך (length) ורוחב (width) בס"מ.

לכל אריח יש מחיר ליחידה (price).

כל משתני המחלקה מוגנים.

עליך להגדיר פונקציה בונה (constructor) המשימה ערך לכל משתנה. הגדר גם פונקציה בונה ריקה.

שים לב שהמשתנה name היינו מצביע. בנה גם פונקציה הורסת (destructor) לאובייקט במחלקה.

ניתן להוסיף (ואף רצוי) פונקציות נוספות למחלקה לפי דרישת התוכנית.

כתוב פונקציה השייכת למחלקה, אשר מקבלת כפרמטרים שני מספרים שלמים המייצגים אורך ורוחב בס"מ של שטח מלבני אשר זקוק לריצוף. הפונקציה תחזיר את מספר האריחים המינימלי הדרוש לריצוף השטח המבוקש מסוג האריח באובייקט עליו היא מופעלת.

המספר המוחזר צריך להיות מספר שלם.

לא ניתן להשתמש באריח מסוים יותר מפעם אחת. כלומר, אם משתמשים רק בחלק מאריח, החלק הנותר מהאריח נזרק

ולא ניתן להשתמש בו שוב.

דוגמא: נניח כי השטח המבוקש הוא בעל המידות 100*100, ומידות האריח הן 20*30 במקרה זה מספר האריחים יהיה 20 (4*5).

נתון מערך מסוג balata המכיל סוגים שונים של אריחים. כתוב פונקציה אשר מקבלת כפרמטרים מערך זה ומספר

האיברים במערך, בנוסף הפונקציה תקבל מידות של שטח מלבני. הפונקציה תדפיס את שם האריח אשר שימוש בו יהווה

את המחיר המינימלי הדרוש לריצוף השטח. הפונקציה תדפיס גם את מספר האריחים הדרוש ואת מחירם הכולל.

ניתן להניח כי קיים רק אריח אחד כזה בעל מחיר מינימלי לריצוף השטח.

בהצלחה

```

08 #include <string.h>
09 class balata
10 {
11     char * name;
12     int length;
13     int width;
14     double price;
15 public:
16     balata(void) {name=NULL;length=0;width=0;price=0.0;}
17     balata(const char * Name, int Lenght, int Width, double Price);
18     ~balata(void){delete [] name;} // can delete with no check have to delete
19     int q1(int w, int l);
20     double q2(int w, int l) {return price*q1(w,l);}
21     void showname(void) {cout<<name;}
22 };
23 void q2(balata a[],int Size, int w, int l);
24 void main(void)
25 {
26     balata c("CCC",10,20,10.0);
27     cout << c.q1(80,70) << endl;
28     cout << c.q1(70,80) << endl;
29     cout << c.q1(65,75) << endl;
30     balata a[] =
31     {
32         balata("AAA",10,20,10.0),
33         balata("BBB",10,20,11.0),
34         balata("CCC",10,20,8.5499),
35         balata("DDD",11,22,9.0)
36     };
37     q2(a,4,100,200);
38     q2(a,4,110,200);
39     q2(a,4,88,220);
40     return;
41 }
42 balata::balata(const char * Name, int Length, int Width, double Price)
43 {
44     name = new char[strlen(Name)+1]; // notice the need for new, strlen, and strcpy
45     strcpy(name,Name); // use this for all char* parameters
46     length=Length;
47     width=Width;
48     price=Price;
49     return;
50 }
51 int balata::q1(int w, int l)
52 {
53     int w1,w2,l1,l2,s1,s2;
54     w1=w/width; w2=w/length;
55     if (w%width!=0) w1++;
56     if (w%length!=0) w2++;
57     l1=l/length; l2=l/width;
58     if (l%width!=0) l2++;
59     if (l%length!=0) l1++;
60     s1=w1*l1; s2=w2*l2;
61     if (s1>s2) return s2; else return s1;
62 }
63 void q2(balata a[],int Size, int w, int l)
64 {
65     double min= a[0].q2(w,l);
66     int minp=0;
67     for(int i=1;i<Size;i++)
68     {
69         double m=a[i].q2(w,l);
70         if (m<min)
71         {
72             min=m;
73             minp=i;
74         }
75     }
76     cout << "Name="; a[minp].showname();
77     cout << " Number="<< a[minp].q1(w,l)<< " Cost="<< a[minp].q2(w,l)<< endl;
78     return;
79 }

```

```

MS-DOS balata
Auto
28
28
28
Name=CCC Number=100 Cost=854.99
Name=DDD Number=95 Cost=855
Name=DDD Number=80 Cost=720
Press any key to continue

```