

יום ה' 5 ספטמבר 2002  
משך המבחן 2 שעות, כל חומר כתוב מותר בשימוש

נתונה המחלקה באה:

```

01 class kuku // another name might have been more appropriate but less fun
02 {
03 private:
04     Int * data; // lots of integers from 0 to 255
05     int size; // number of integers in data
06     int sign; // can you guess what I do? I can hold the values -1 or 1
07
08 public:
09     kuku(void); // default constructor
10     ~kuku(void); // destructor
11     kuku(const int * Data, int Size, int Sign); // constructor
12     kuku(const kuku & rhs); // copy constructor
13 };

```

כפי שניתן לראות למחלקה מוגדרים פונקציה בונה של ברירת המחדל (Default Constructor), פונקציה הורסת (Destructor), פונקציה בונה להעתקה (Copy Constructor) ופונקציה בונה נוספת מערך של שלמים, סימן, ומספר שלם שהוא מספר האיברים במערך השלמים ומאתחלת את המחלקה בהתאם.

- עבור 40 נקודות כתוב את הפונקציות המוגדרות ותוכנית ראשית שתדאג להפעיל את כולן ותדגים גם את עבודת הפונקציה המבוקשת בשאלה השלישית ואת אופרטור השוויון המבוקש בשאלה השנייה.
- שני קוקויים מוגדרים כשווים אם הסימן זהה, מספר האיברים במערך השלמים זהה, וכל אחד מהאיברים במערך השלמים של האחד זהה לאבר המקביל של השני. עבור 30 נקודות הגדר וכתוב אופרטור השוואה (==) למחלקה.
- עבור 30 נקודות כתוב פונקציה למחלקה מחזירה משתנה מסוג long. ערכו של ה long יחושב כמכפלת הסימן בסכום האיברים הבאים:  
שלם האחרון במערך השלמים.  
מכפלת השלם לפני האחרון במערך השלמים ב 2  
מכפלת השלם לפני השלם לפני האחרון ב 4  
מכפלת השלם לפני ..... ב 8  
וכן הלאה מכפלה בחזקות של 2 עד לשלם הראשון במערך השלמים  
לדוגמה אם מערך השלמים מכיל את הערכים הבאים:  
2, 10, 100, 10, 1, 7, 12 sign=-1  
אז ערכו של ה long המוחזר יחושב בצורה הבאה:  
 $(-1) * (2 * 64 + 10 * 32 + 100 * 16 + 10 * 8 + 1 * 4 + 7 * 2 + 12)$

בהצלחה

בני

```

01 #include <iostream.h>
02 class kuku // another name might have been more appropriate but less fun
03 {
04 private:
05     int * data; // lots of integers
06     int size; // number of integers in data
07     int sign; // can you guess what I do? I can hold the values -1 or 1
08     operator=(const kuku & rhs); // can't have one
09 public:
10     kuku(void); // default constructor
11     ~kuku(void); // destructor
12     kuku(int * Data, int Size, int Sign); // constructor
13     kuku(const kuku & rhs); // copy constructor
14     bool operator==( const kuku & rhs); // for q2
15     long q3(void);
16     friend ostream & operator<<(ostream & lhs, const kuku & rhs);
17 };
18 void main(void)
19 {
20     kuku k;                                cout << k << endl;
21     int temp[]={2,10,100,10,1,7,12};
22     kuku k1(temp,7,-1);                    cout << k1 << endl;
23     kuku k2(k1);                          cout << k2 << endl;
24     cout << (k2==k1) << endl;              cout << (k2==k) << endl;
25     cout << k1.q3() << endl;
26     return;
27 }
28 kuku::kuku(void) // default constructor
29 {
30     size=0; sign=1; data=NULL;
31 }
32 kuku::~kuku(void) // destructor
33 {
34     delete [] data;
35 }
36 kuku::kuku(int * Data, int Size, int Sign) // constructor
37 {
38     data=new int[Size];
39     size=Size;                                sign=Sign;
40     for (int i=0;i<size;i++) data[i]=Data[i];
41 }
42 kuku::kuku(const kuku & rhs) // copy constructor
43 {
44     size=rhs.size;                            sign=rhs.sign;
45     data=new int[size];
46     for (int i=0;i<size;i++) data[i]=rhs.data[i];
47 }
48 bool kuku::operator==( const kuku & rhs)
49 {
50     if (size==rhs.size && sign==rhs.sign)
51     {
52         for(int i=0;i<size;i++)
53             if (data[i]!=rhs.data[i]) return false;
54         return true;
55     }
56     else
57         return false;
58 }
59 long kuku::q3(void)
60 {
61     long temp=0;                                long t=1;
62     for(int i=0;i<size;i++)
63     {
64         temp+=data[size-1-i]*t;
65         t*=2;
66     }
67     return temp*sign;
68 }
69 ostream & operator<<(ostream & lhs, const kuku & rhs) // not part of the answer
70 {
71     lhs<<"size:"<<rhs.size<<endl;
72     lhs<<"sign:"<<rhs.sign<<endl;
73     for (int i=0;i<rhs.size;i++) lhs<<(int)rhs.data[i]<<" ";
74     return lhs;
75 }

```