

Задача 4 «Зернинки»

Задача 5. Банкомат

Задачі 6. Купування квитків

Задача 4 «Зернинки»

Мишка збирає зернинки на шаховій дошці. Може рухатись вниз та вліво. Зібрати найбільше зернинок.

1

2

3

4

5

6

7

8

9

1

2

3

4

5

6

7

8

9

1

2

1

0

8

4

1

4

9

6

1

2

3

3

12

16

17

20

30

36

2

6

2

5

7

3

2

10

1

3

2

8

10

15

22

25

27

37

37

40

3

2

8

6

6

6

6

8

5

8

3

10

17

24

30

35

41

49

55

62

4

5

9

5

9

6

6

4

6

9

4

15

26

31

41

47

53

57

63

72

5

7

4

6

7

3

5

8

4

1

5

21

30

38

48

51

58

66

71

73

6

2

5

4

9

4

5

5

4

10

6

24

35

41

56

60

66

72

75

85

7

4

3

8

1

0

6

6

9

4

7

28

39

49

58

61

71

77

87

91

8

4

3

7

8

8

9

5

9

1

8

32

42

57

65

74

83

89

98

99

9

2

5

2

2

2

1

6

7

6

9

34

47

59

68

76

84

95

105

110

10

10

1

6

2

10

7

4

5

3

10

44

48

65

69

86

93

99

110

113

=МАКС(С3+М3;С3+N2)

Задача 5

Час на тест - 30 секунд.

У деякій державі в обігу знаходяться банкноти певних номіналів. Національний банк

хоче, щоб банкомат видавав будь-яку запитану суму за допомогою мінімального числа банкнот, вважаючи, що запас банкнот кожного номіналу необмежений. Допоможіть Національному банку вирішити цю задачу.

Формат вхідних даних

Перший рядок вхідних даних містить натуральне число n , $0 < n \leq 100$. Другий рядок вхідних даних містить n різних натуральних чисел $x_1, x_2, \dots, x_n$, не переважаючих 1.000.000. Третій рядок містить натуральне число S , не перевершує 5.000.000.

Формат вихідних даних

Програма повинна знайти представлення числа S у вигляді суми доданків з множини $\{x_i\}$, що містить мінімальне число доданків і вивести це подання на екран (у вигляді послідовності чисел, розділених пробілами). Якщо таких уявлень існує декілька, то програма повинна вивести будь-яке (одне) з них. Якщо таке подання не існує, то програма повинна вивести слово impossible.

Приклад

Вхідні дані

5

1 3 7 12 32

40

Вихідні дані

1 7 32

Вхідні дані

5

10 50 100 500 1000

99

Вихідні дані

Impossible

```
#include <cstdlib>
```

```
#include "iostream"
```

```
#include "fstream"
```

```
using namespace std;
```

```
int main ()
```

```
{ifstream cin ("input0.txt");
```

```
ofstream cout("output0.txt");
```

```
int n,k,a[100],b[100];
```

```
int m, i;
```

```
cin>>k;
```

```
for (i=0;i<k;i++)cin>>a[i];
```

```
cin>>n;
```

```
//cout<<"n="<<n<<endl;for (i=0;i<k;i++)cout<<a[i]<<endl;
```

```
const int INF=1000000000;// Значення константи "нескінченність"
```

```
int F[1000];
```

F[0]=0;

for(m=1;m<=n;++m) // заповнюємо масив A

{ // m - сума, котору видати

F[m]=INF; // відмічаємо, що суму i видати не можна

for(i=0;i<k;++i) // перебираємо всі номінали банкнот

{

if(m>=a[i] && F[m-a[i]]+1<F[m])

F[m] = F[m-a[i]]+1; // змінюємо значення F[m], якщо знайшли

} // кращий спосіб видати суму m

}

```
int j=0;
```

```
if (F[n]==INF)
```

```
cout<<"Impossible"<<endl;
```

```
else
```

```
{
```

```
while(n>0)
```

```
{
```

```
for(i=0;i<k;++i)
```

```
{
```

```
if(F[n-a[i]]==F[n]-1)
```

```
{
```

```
j++;
```

```
b[j]=a[i];
```

```
n=n-a[i];
```

```
break;
```

```
}}}}
```

```
for (i=1;i<j;i++)cout<<b[i]<<" "; cout<<b[j]<<endl; }
```

Задачі 6. Купування квитків

Ім'я файлу програми:

Bilet.*

Ім'я вхідного файлу:

Bilet.dat

Ім'я вихідного файлу:

Bilet.dat

Максимальний час роботи на одному тесті:

5 секунд

Максимальна оцінка за завдання:

100 балів

За квитками на прем'єру нового мюзиклу вишикувалася черга з N людей, кожний з яких хоче купити 1 квиток. На всю чергу працювала тільки одна каса, тому продаж квитків йшов дуже повільно, приводячи людей черги у відчай. Найкмітливіші швидко відмітили, що, як правило, декілька квитків в одні руки касир продає швидше, ніж коли ці ж квитки продаються поодинці. Тому вони запропонували декільком людям, які стоять підряд віддавати гроші першому з них, щоб він купив квитки на всіх.

Проте для боротьби із спекулянтами касир продавала не більше 3-х квитків в одні руки, тому домовитися таким чином між собою могли лише 2 або 3 підряд вартих людини.

Відомо, що на продаж i -їй людині з черги одного квитка касир витрачає A_i секунд, на продаж двох квитків - B_i секунд, трьох квитків - C_i секунд. Напишіть програму, яка підрахує мінімальний час, за який могли бути продані квитки для всіх людей черги.

Добавил(a) Administrator
30.05.13 15:12 -

Зверніть увагу, що квитки на групу людей, що об'єдналися, завжди купує перший з них. Також ніхто в цілях прискорення не купує зайвих квитків (тобто квитків, які нікому не потрібні).

Формат вхідних даних

У вхідному файлі записано спочатку число N - кількість покупців в черзі ($1 \leq N \leq 5000$). Далі йде N трійок натуральних чисел A_i, B_i, C_i . Кожне з цих чисел не перевищує 3600. Люди в черзі нумеруються починаючи від каси.

Формат вихідних даних

У вихідний файл виведіть одне число - мінімальний час в секундах, за яке могли бути обслужені всі покупці.

Приклади

Bilet.dat

Bilet.sol

5

5 10 15

2 10 15

5 5 5

20 20 1

20 1 1

12

2

3 4 5

1 1 1

4

$N=5$

i

Ai

Bi

Ci

1

5

10

15

2

2

10

15

3

5

5

5

4

20

20

1

5

20

1

1

$$D[i] = \min (D[i-1] + A_i, D[i-2] + B_{i-1}, D[i-3] + C_{i-2})$$

D[1]

D[2]

D[3]

D[4]

D[5]

5

7

5

6

12 - відповідь завдання

$d[0] = 0;$

$d[1] = a[1];$

$d[2] = \min(a[1] + a[2], b[1]);$

for ($i=3; i \leq n; i++$) $d[i] = \min(d[i-1] + a[i], \min(d[i-2] + b[i-1], d[i-3] + c[i-2]));$