



Олимпиада
«Технологическое
предпринимательство»

Наименование образовательного учреждения: ФГБОУ ВО «Сибирский государственный аэрокосмический университет имени академика М.Ф. Решетнева»

Очный этап Олимпиады «Технологическое предпринимательство»

Бланк заполняется печатными буквами

Олимпиада по ИНФОРМАЦИОННЫМ ТЕХНОЛОГИЯМ класс 9
Фамилия ФЕДОРЕНКО
Имя ИЛЬЯ
Отчество АЛЕКСЕЕВИЧ
Дата рождения 23.11.2000
Страна РОССИЯ
Регион КРАСНОЯРСКИЙ КРАЙ

Заполняется организатором

Задание №1

Можно заметить, что символы данного алфавита — это арабские цифры соединенные с их зеркальным отражением:

$$\uparrow \rightarrow 1 + \uparrow$$

$$\text{S} \rightarrow 2 + \text{S}$$

$$\text{A} \rightarrow 4 + \text{A}$$

$$\text{A} \rightarrow 7 + \text{A}$$

Запишем весь алфавит, отобразив все оставшиеся цифры:

∞	↑	∩	⊖	⊕	⊞	⊟	⊠	⊡	⊢
0	1	2	3	4	5	6	7	8	9

Задание №2

Язык программирования: python 3.6.

```

n = int(input()) # ввод данных
sum = [100] # массив сумм, пока за первый шаг
for i in range(n-1): # цикл, проходящий n-1 раз (n-1, т.к. первое уже есть)
    sum.append(sum[i]*2) # добавляет в массив еще не округленное
                        # число следующей суммы.
while len(str(sum[i+1]))/2 > str(sum[i+1]).count('0'):
    # число заходит в этот цикл, если оно не является "круглым"
    # для этого считаем длину числа и делим её на 2, далее ищем
    # количество нулей при помощи метода count
    sum[i+1] += 10** (str(sum[i+1]).count('0'))
    # прибавим, пока число не станет достаточно "круглым"
print('\n'.join(sum)) # выводим ответ, используем метод join.

```

Задание №3

Язык программирования: python 3.6.

```
h = int(input()) # количество ячеек
w = int(input())
pole = []
```

```
for i in range(h):
```

```
    pole.append([i for i in input().split()])
```

```
kod = 0 # нулевой ход - это время в сере секундах
```

```
for i in range(h):
```

```
    for j in range(w):
```

```
        if pole[i][j] == 'R': # заменили положение робота 0.
            pole[i][j] = 0
            break
```

```
while True:
```

```
    kod += 1
```

```
while f == True:
```

```
    kod += 1
```

```
    f = False
```

```
# если мы найдем ячейку со значением kod-1, то цикл завершится
```

```
for i in range(h):
```

```
    for j in range(w):
```

```
        if pole[i][j] == kod - 1:
```

```
            f = True
```

```
            if i != 0:
```

```
                # чтобы не выйти за границы массива (города)
```

```
                    if pole[i-1][j] == '.':
```

```
                        pole[i-1][j] = kod
```

```
                    elif pole[i-1][j] == 'S':
```

```
                        break
```

```
# если мы прошли до точки
```

```
            if i != h:
```

```
                if pole[i+1][j] == '.':
```

```
                    pole[i+1][j] = kod
```

```
                elif pole[i+1][j] == 'S':
```

```
                    break
```

```
            if j != 0:
```

```
                if pole[i][j-1] == '.':
```

```
                    pole[i][j-1] = kod
```

```
                elif pole[i][j-1] == 'S':
```

```
                    break
```

Задание №3

```
if j != w:  
    if pole[i][j+1] == '.':  
        pole[i][j+1] = hod  
    elif pole[i][j+1] == 's':  
        break
```

```
if f == True:  
    print(hod)  
else:  
    print(-1)
```

если узел прервался, то $f == True$, значит, робот
добрался до цели на ходу номер hod
если узел завершился, то $f == False$, значит,
робот не добрался до цели

Примечание: в первом примере робот сможет добраться до
цели за 4 хода

Решение верное, но не оптимальное.

Попробуйте посмотреть оптимизацию таких задач на графах.

Задание №4

```

from math import ceil
x, n = [int(i) for i in input().split()] # ввод и считывание данных
s, w = [int(i) for i in input().split()]
q, m = [int(i) for i in input().split()]

k = w / m

```

```

if k == 1: # ситуация, когда компьютер обрабатывает столько же, сколько
    st = s + q # имеет программист.
    if st * ceil(n/w) <= x:
        print(st * ceil(n/w))
        print(n/w)
        print(ceil(n/w), ceil(n/w))
        print(1)

```

Рассмотрен лишь 1 пример.