

④ 5

кол-во исправленных членов: 3

Был на лекции $n: 2^1 < 3 < 2^2 \Rightarrow n=2$

1. инф. объём страницы: $P = \frac{11 \cdot 400 \cdot 950 \text{ бит/байт}}{8 \text{ байт}} = \frac{4.200.950 \text{ байт}}{8} = 525.118,75 \text{ байт}$

2. размер стр. с текстом: $T = \frac{15 \text{ байт} \cdot 449 \text{ байт}}{8 \text{ байт}} = 898 \text{ байт}$

3. X - канал сдр. с рисунками, Y - с тактом

По условию, текстовые стр. занимают в 30 раз меньше памяти

Суммарный объем страны равен $1,5 \text{ Мбайт} = 1,5 \cdot 1024 \cdot 1024 \text{ байт}$

$$\begin{cases} \frac{x \cdot P}{y \cdot T} = 30 \\ P \cdot x + y \cdot T = 1,5 \cdot 2^{20} \text{ \textcircled{B}}_{\text{ant}} \end{cases} \quad \begin{cases} x \cdot P = 30 \cdot y \cdot T \\ 30 \cdot y \cdot T + y \cdot T = 1,5 \cdot 2^{20} \text{ \textcircled{B}}_{\text{ant}} \end{cases} \quad \begin{cases} x = \frac{30 \cdot y \cdot T}{P} \\ y = \frac{1,5 \cdot 2^{20} \text{ \textcircled{B}}_{\text{ant}}}{31 \cdot T} \end{cases}$$

$$\begin{cases} y = \frac{15 \cdot 2^{10} \text{ \AA}}{31.899 \text{ \AA}} \\ x = \frac{30 \cdot 15 \cdot 2^{10} \text{ \AA}}{31.95000 \text{ \AA}} \end{cases}$$

$$x + y = \frac{45 \cdot 2^{10}}{31} \left(\frac{1}{898} + \frac{3}{9500} \right) \approx \frac{72}{73}$$

Amber: 72

⑤ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 / var. cotozine nascenta
 $n=20, M = \frac{A[20]}{2} = 10$
 пока $A[i] - 1 \geq M$

-1, Таким образом элемент с индексом из 20 будет глупо-ся влево, пока становится левее
 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
 1 2 3 4 5 6

- 1-е первое выполнение внутр. гоним

[illegible]

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
при первом 4 2 секунд к убер. 20 10

2-3: ha 11

4-5: 12

6-7:13

8-9:14

Смешанное, набухает $k = 120$

One beam: 120

Стр. 1

```
#include <iostream>
#include <string>
using namespace std;
```

```
int main() {
    byte m[32]; # память
```

```
    int ptr=0; # указатель исп. для заполнения памяти
    int cycles[32]; # заполним кол-во перезаписей ячеек зн. 0
    for (int i=0; i<32; i++)
        cycles[i]=0;
```

```
    int state=0; # опр. ожидаемый ввод
    int rem=1000; # кол-во введ. перезаписей
    while (true) { # в. цикл
```

```
        if (state==0) {
            string s; cin>>s;
```

```
            if (s=="add") {
                cout<<"Введите время издержек в часах";
                state=1; # переход в сост. ожидание ввода часа
```

```
            }
            if (s=="all") { # всего в 32 ячейки попросится 10 записей по 3 значения (так как в 32 ячейки можно поместить 10х3 зн.)
                for (int i=0; i<30; i+=3) { if (cycles[(ptr+i)%32]>0) { # пропустить блок 9 часов
                    cout<<m[(ptr+i)%32]<<" "<<m[(ptr+i+1)%32]<<" "<<m[(ptr+i+2)%32]<<" T="<<(int)(m[(ptr+i)%32]-50)<<endl;
                } } # вывод значений только тех ячеек, которые перезаписаны хотя раз
```

```
            }
            if (s=="loss") {
```

```
                cout<<"В оперативную память можно вписать "<<rem<<" издержек"<<endl;
```

```
            }
            if (s=="cycles") {
```

```
                state=4;
                cout<<"Введите номер ячейки"<<endl;
```

```
            }
        }
        if (state==1) { if (rem==0) { cout<<"Ошибка: невозможно перезаписать ячейку!"; state=0; continue; }
            cin>>m[ptr++];
            state=2;
            cout<<"Введите время издержек в минуту"<<endl; # ввод соотв. значения памяти,
            rem--; # перем. указателя, изм-е кол-ва
            cycles[ptr-1]++; # год. перезаписей
```

```
        }
        if (state==2) { if (rem==0) { cout<<"Ошибка: невозможно перезаписать ячейку!"; state=0; continue; }
            cycles[ptr]++; # аналогично state==1 и state==3
            rem--;
            cin>>m[ptr];
            ptr++;
            state=3;
            cout<<"Время издержек"<<m[ptr-1]<<" "<<m[ptr]<<" "<<m[ptr+1]<<" Введите значение температуры в градусах Цельсия"<<endl;
```

```
        }
        if (state==3) { if (rem==0) { cout<<"Ошибка: невозможно перезаписать ячейку!"; state=0; continue; }
            cycles[ptr]++;
            rem--; int val; cin>>val;
            m[ptr]=val+50;
            state=0;
            cout<<"Добавлено издержек: B"<<m[ptr-1]<<" "<<m[ptr]<<" "<<m[ptr+1]<<" T="<<val<<endl;
            if (ptr==31) { ptr=0; } else { ptr++; }
        }
        if (state==4) {
            int i; cin>>i;
            cout<<"Ячейка"<<i<<" была перезаписана"<<cycles[i]<<" раз"<<endl;
            state=0;
        }
    }
    return 0;
```

СТР. 2


```
if state == 0 and s == 'show': # вывод матрицы
```

```
for i in range(0, 8):
```

```
for j in range(0, 8):
```

```
print(int(mat[i][j] + 0.5), end=" ") # округление зн. ячейки до целого по правилам арифметики
```

```
print("") # переход на новую строку
```

Стр. 7

