

Лабораторная работа №6

Разложение чисел на множители

Кубасов В.Ю.

Вводная часть

- Вычисление элементарных операций сложения вне пределов размещающихся в памяти целочисленных типов необходимо при работе с большими числами

Реализовать алгоритмы:

- сложения
- вычитания
- умножения
- деления

Выполнение работы

Сложение

```
def add(u = "", w = "", b = 10):  
    k = 0  
    ans = ""  
  
    for i in range(len(u))[::-1]:  
        ans = chr((ord(u[i]) + ord(w[i]) - 2 * ord('0') + k) % b + ord('0'))  
        k = int((ord(u[i]) + ord(w[i]) - 2 * ord('0') + k) / b)  
  
    if (k > 0):  
        ans = '1' + ans  
    print(ans)  
  
add("9824211231231232131", "9824211231231232131")
```

```
def sub(u = "", w = "", b = 10):  
    k = 0  
    ans = ""  
  
    for i in range(len(u))[::-1]:  
        ans = chr(abs(ord(u[i]) - ord(w[i]) + k) % b + ord('0')) + ans  
        k = int((ord(u[i]) - ord(w[i]) + k) / b)  
  
    print(ans)  
  
sub("9824211231231232131", "9824211231231232131")
```

Умножение

```
ZERO = ord('0')
```

```
def mul(u = "", v = "", b = 10):
```

```
    k = 0
```

```
    ans = [0] * (len(u) + len(v))
```

```
    for j in range(len(v))[::-1]:
```

```
        if (v[j] == 0):
```

```
            ans[j] = 0
```

```
        else:
```

```
            k = 0
```

```
            for i in range(len(u))[::-1]:
```

```
                t = (ord(v[j]) - ZERO) * (ord(u[i]) - ZERO) + ans[i + j]
```

```
                ans[i + j] = t % b
```


Умножение быстрое

```
def mulf(u = "", v = "", b = 10):  
    t = 0  
    ans = [0] * (len(u) + len(v))  
  
    for s in range(len(ans) - 1):  
        for i in range(s):  
            t = (ord(u[len(u) - 1 - i]) - ord('0')) * (ord(v[len(v) - 1 - s]) - ord('0')) + t  
            ans[len(v) + len(u) - s] = t % b  
            t = int(t / b)  
  
    print(''.join(str(x) for x in ans))  
  
mulf("9824211231231232131", "9824211231231232131")
```

Деление с утечкой памяти

```
ZERO = ord('0')
```

```
def div(u = "", v = "", b = 10):
```

```
    t = 0
```

```
    q = [0] * (len(u) - len(v))
```

```
    r = [0] * len(v)
```

```
    while(u >= v * b ** (len(u) - len(v))):
```

```
        for i in range(len(v) + 1, len(u) + 1)[::-1]:
```

```
            if (u[i] > v[0]):
```

```
                q[i - len(v) - 1] = b - 1
```

```
            else:
```

```
                q[i - len(v) - 1] = int((u[i] * b + u[i - 1]) / v[0])
```

Выводы:

Выводы:

- В ходе работы реализовали предложенные алгоритмы