

Вариант 2:

Нека е даден следния шаблон на структура, дефинираща елемент от едносвързан списък:

```
template <typename T>
struct box {T data; box *next;};
```

Задача 1. Да се дефинира подходящо параметризирана функция `fillGaps(L)`, където `L` е указател към първия елемент на линеен едносвързан списък, чиито елементи са цели числа в намаляващ ред ред. Между всеки два съседни елемента `K` и `L` на списъка функцията да създава и вмъква последователно всички елементи от редицата $K/2, K/4, K/8, \dots$, които са по-големи от `L`, ако има такива.

Пример: Списъкът $256 \rightarrow 32 \rightarrow 16 \rightarrow 1$, ще се преобразува до $256 \rightarrow \underline{128} \rightarrow \underline{64} \rightarrow 32 \rightarrow 16 \rightarrow \underline{8} \rightarrow \underline{4} \rightarrow \underline{2} \rightarrow 1$. Подчертани са новите елементи.

Задача 2. Да се дефинира функция `void keepProducts ([подходящ тип] first)`, където `first` е указател към първия елемент на едносвързан списък `L` с елементи a_0, \dots, a_n . От списъка да се премахнат всички елементи (освен първия и втория), които не са кратни на предишните два елемента.

Пример: Списъкът $2, 3, \underline{3}, 12, 72, \underline{36}$ ще се измени до $2, 3, 12, 72$. Подчертани са елементите, които са изтрети.

Задача 3. Да се дефинира функция `void mirror ([подходящ тип] first)`. Функцията да обръща прилепя към едносвързания списък с указател към първия елемент `first` негово огледално копие.

Пример: Ако списъкът съдържа елементите $1, 2, 3, 4, 5$ в този ред, то той ще се измени до $1, 2, 3, 4, 5, 5, 4, 3, 2, 1$.

Време за работа: 100 минути