

2.1 ОДНОСВЯЗНЫЕ ЛИНЕЙНЫЕ СПИСКИ.

Постановка задачи

Обеспечить пользователю возможность ввода данных, при необходимости обеспечить контроль ввода, при добавлении элементов списка реализовать **добавление в конец списка**. При вводе списков должна быть предусмотрена возможность ввода **пустого списка**, с которым алгоритмы так же должны корректно обрабатывать. Выполнить обработку данных в соответствии с заданием. Если в задании явно не указано на необходимость формирования нового списка, то в процессе решения требуется модифицировать исходный список. Вывести результаты обработки на экран. После вывода результатов, предоставить пользователю возможность завершить работу с программой или продолжить работу с новыми данными. Обеспечить освобождение выделенной в процессе работы программы динамической памяти. Все алгоритмические подзадачи (добавление элемента списка, удаление элемента списка, вывод списка на экран, очистка списка, ввод элементов списка и т.д.) выделять в соответствующие подпрограммы.

Односвязные линейные списки

Односвязный линейный список является простейшим типом динамической структуры данных, состоящей из узлов. Каждый узел включает в себя в классическом варианте два поля:

- Поле данных – хранимые данные, которые могут быть представлены переменные базовых и производных типов, в том числе структур.
- Поле связи – указатель на следующий узел в списке, причем поле связи последнего узла списка содержит нулевое значение (указывает на NULL).

Элементы связанного списка можно помещать и исключать произвольным образом. Доступ к списку осуществляется через указатель, который содержит адрес первого элемента списка, называемый **головой** (корнем) списка.



При добавлении нового узла в такую структуру выделяется новый блок памяти, заполняется поле данных и устанавливаются связи нового элемента с уже существующими. При удалении элемента нет необходимости сдвигать все элементы, следующие за удаляемым, достаточно перебросить соответствующие указатели и освободить блок памяти, занимаемый удаляемым элементом. Перемещение по списку организуется с помощью указателя, который изначально инициализируется головой списка, до тех пор, пока он не будет равен NULL.

Пример реализации

Ниже приведены описание структуры данных, для хранения узлов односвязного линейного списка целых чисел, а также варианты реализации процедур добавления узла в голову списка и вывода списка на экран:

```
struct node
{
    int data; // поле данных
    struct node *next; // указатель на следующий элемент
};
```

```

// добавление узла в голову списка
void add_first(node **head, int data)
{
    node *temp;
    temp = (node*)malloc(sizeof(node));
    temp->data = data; // сохранение поля данных добавляемого узла
    temp->next = *head; // инициализация поля ссылки
    *head = temp;
}

// вывод списка на печать
void print_list(node *head)
{
    while (head)
    {
        printf("[%d]->", head->data);
        head = head->next;
    }
    printf("NULL\n");
}

```

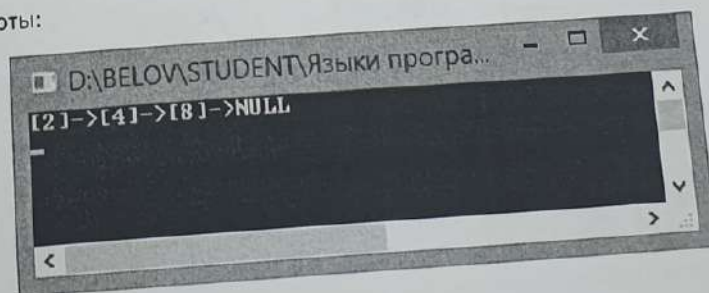
Вызов процедур. Создание и печать списка:

```

...
node * head = NULL;
add_first(&head, 8);
add_first(&head, 4);
add_first(&head, 2);
print_list(head);
...

```

Результат работы:



Варианты:

1. Даны два списка действительных чисел. Сформировать третий список так, что каждый его элемент является максимумом из соответствующих значений первых двух списков, причем если значения равны, то в результирующий список элемент не добавляется. Если длина списков различна, то недостающие значения заменять нулями.
2. Дан список вещественных чисел. Сформировать новый список вещественных чисел такой же размерности, такой, что каждый его i -й элемент ($i = 0..N-1$) является максимальным, среди первых $i + 1$ элементов исходного списка.
3. Даны два списка целых чисел. Сформировать третий список как симметрическую разность первых двух списков (элементы принадлежащие только первому или только второму списку). Элементы, встречающиеся в исходных списках более одного раза должны попасть в результирующий список однократно.
4. Дан список двумерных точек с целочисленными координатами. Оставить для каждого элемента в списке не более чем N его первых вхождений. Значение $N \geq 1$ задаёт пользователь.
5. Даны два списка действительных чисел. Объединить два списка в третий так, чтобы элементы на нечётных позициях были из первого списка, а на чётных — из второго. Если длина списков различна, то недостающие элементы в результирующем списке заменять нулями.
6. Даны два списка действительных чисел. Заменить каждый N -й элемент первого списка на первые K элементов второго списка. Значения N и K задаёт пользователь.
7. Дан список, состоящий из букв русского алфавита и символов-разделителей. Заменить каждую последовательность символов разделителей в списке на один случайный символ-разделитель из этой последовательности.
8. Дан список двумерных точек. Вставить после каждой N -ой точки K её копий. Значения N и K задаёт пользователь.
9. Дан список слов, составленных из букв русского алфавита. Все слова в списке, удовлетворяющие заданной маске (маска содержит буквы и символ $*$ который заменяет любую букву) заменить на заданное пользователем слово. Сравнение символов не должно зависеть от регистра.
10. Даны два списка целых чисел, вставить после каждого элемента первого списка такие элементы из второго, которые больше либо равны ему. При вставке элементов сохранить их порядок следования во втором списке.
11. Даны два списка целых чисел. Найти в первом списке наименьший и наибольший элемент и заменить все вхождения наименьшего элемента на элементы второго списка в прямом направлении, а все вхождения наибольшего элемента на элементы второго списка в обратном направлении.
12. Даны два списка целых чисел. Вывести в третий список пересечение первых двух списков (элементы принадлежащие и первому и второму списку). Элементы, встречающиеся в исходных списках более одного раза должны попасть в результирующий список однократно.
13. Даны два списка целых чисел. Каждый четный элемент первого списка заменить на элементы второго списка которые меньше него, а каждый нечетный — на элементы которые больше него. При вставке элементов сохранить их порядок следования во втором списке.

14. Дан список целых чисел. Заменить в исходном списке подряд идущие одинаковые элементы одним.
15. Дан список вещественных чисел – список коэффициентов многочлена $f(x)$. Сформировать списки коэффициентов всех возможных производных этого многочлена. Рассчитать значение $f(x)$ и всех его производных в заданной точке.
16. Дан список двумерных точек с вещественными координатами. Упорядочить список в порядке возрастания расстояния до заданной пользователем точки.
17. Дан список символов. Сформировать два списка: список символов и список целых чисел. В список символов поместить все символы без повторов, присутствующие в исходном списке, в порядке их появления, а в список целых – количество повторов этих символов в исходном списке.
18. Дан список действительных чисел. Заменить в исходном списке каждые N элементов их суммой, если число элементов списка не кратно N , то последнюю сумму составят оставшиеся $M < N$ элементов. Значение N задаёт пользователь.
19. Даны два списка действительных чисел. Заменить каждый элемент первого списка, принадлежащий ϵ -окрестности заданного пользователем элемента, на элементы второго списка. Значение ϵ задаёт пользователь.
20. Даны два списка действительных чисел. Сформировать третий список так, что каждый его элемент является суммой соответствующих элементов первых двух списков. Если длина списков различна, то недостающие слагаемые заменять нулями.
21. Дан список букв русского алфавита. Все последовательности в списке, удовлетворяющие заданной строке-маске (содержит буквы и символ $*$ который заменяет любую букву) заменить на заданный пользователем символ. Сравнение символов не должно зависеть от регистра.
22. Даны два списка целых чисел. Сформировать третий список как объединение первых двух списков (элементы принадлежащие или первому, или второму списку). Элементы, встречающиеся в исходных списках более одного раза должны попасть в результирующий список однократно.
23. Даны два списка список букв русского алфавита. Все вхождения заданной пользователем строки в первом списке заменить элементами второго списка. Сравнение символов не должно зависеть от регистра.
24. Даны два списка строк. Сформировать третий список так, что каждый его элемент является конкатенацией соответствующих элементов первых двух списков, причем для чётных элементов строки конкатенируются по правилу $a+b$, для нечетных $b+a$. Если длина списков различна, то недостающие строки заменять пустой строкой.
25. Даны два списка строк. Сформировать третий список так, что каждый его элемент является строкой максимальной длины среди соответствующих элементов первых двух списков, причем если длины строк равны, то в результирующий список элемент не добавляется. Если длина списков различна, то недостающие строки заменять пустой строкой.
26. Даны два списка действительных чисел. Заменить все вхождения второго списка в первом на заданное пользователем значение.
27. Дан текст, содержащий буквы русского алфавита и символы разделители. Преобразовать его в список слов, не используя стандартную функцию поиска лексем в строке. Упорядочить список по возрастанию длины слов.

28. Дан
списке э
нечетн
не д

28. Даны два списка букв русского алфавита. Каждое четное вхождение заданной буквы в списке заменять на последовательность чётных элементов второго списка, соответственно каждое нечетное вхождение заменять на последовательность нечётных элементов. Сравнение символов не должно зависеть от регистра.

29. Даны два списка действительных чисел. Заменить каждый элемент первого списка, на элементы второго списка, принадлежащие его ϵ -окрестности. При вставке элементов сохранить их порядок следования во втором списке. Значение ϵ задаёт пользователь.