МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ государственное БЮДЖЕТНОЕ

образовательное учреждение

высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра защиты информации

**

**Лабораторная работа №7**

**«**Рекурсивные функции»

**по дисциплине: «*Программирование*»**

|  |  |
| --- | --- |
| *Выполнил:*  *Студент гр. «АБ-324», «АВТФ»*  *Тятюшкин М.О.*  *«\_» апреля 2024г*  *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*  *(подпись)* | *Проверил:*  *Ассистент кафедры ЗИ*  *Исаев Глеб Андреевич*  *«\_\_\_» \_\_\_\_\_\_ 2024г*  *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*  *(подпись)* |

Новосибирск 2024

**Цели и задачи работы**: изучение рекурсивного программирования, методов разработки эффективных алгоритмов.

**Задание к работе**: Реализовать циклический вычислительный процесс. Самостоятельно решить задачи в соответствии с индивидуальным вариантом.

**Задания:**

**Задание 1.** Имеется N контейнеров высоты H. Задано множество предметов, каждый из которых имеет свою высоту. Можно ли разместить предметы в этих контейнерах так, чтобы груз не выступал над контейнером?

**Задание 2.** Дана закодированная строка, верните её декодированную версию. Правило кодирования: k[encoded\_string], где encoded\_string квадратные скобки повторяются ровно k раз. Обратите внимание, что k это гарантированно положительное целое число. Вы можете предположить, что входная строка всегда действительна; нет лишних пробелов, правильные квадратные скобки и т.д. Кроме того, вы можете предположить, что исходные данные не содержат никаких цифр и что цифры предназначены только для тех повторяющихся чисел k. Например, не будет ввода типа 3a или 2[4].

**Методика выполнения работы**:

1) Изучить технологию использования стека при рекурсивном программировании.

2) Написать рекурсивную программу решения поставленной задачи.

3) Протестировать программу.

**Задания**

# Задание 1. Имеется N контейнеров высоты H. Задано множество предметов, каждый из которых имеет свою высоту. Можно ли разместить предметы в этих контейнерах так, чтобы груз не выступал над контейнером?

## C++:

#include <iostream>

#include <vector>

using namespace std;

bool canFit(vector<int>&containers, vector<int>&items, int containerIndex) {

if (containerIndex == containers.size()) {

return true; // Все контейнеры обработаны успешно

}

for (int i = 0; i < items.size(); ++i) {

if (items[i] <= containers[containerIndex]) {

int itemHeight = items[i];

containers[containerIndex] -= itemHeight;

if (canFit(containers, items, containerIndex + 1)) {

return true;

}

containers[containerIndex] += itemHeight; // Откат изменений

}

}

return false;

}

int main() {

setlocale(LC\_ALL, "Rus");

int N;

cout << "Введите количество контейнеров: ";

cin >> N;

vector<int> containers(N);

cout << "Введите высоту каждого контейнера:\n";

for (int i = 0; i < N; ++i) {

cin >> containers[i];

}

int M;

cout << "Введите количество предметов: ";

cin >> M;

vector<int> items(M);

cout << "Введите высоту каждого предмета:\n";

for (int i = 0; i < M; ++i) {

cin >> items[i];

}

if (canFit(containers, items, 0)) {

cout << "Предметы можно разместить в контейнерах так\n";

}

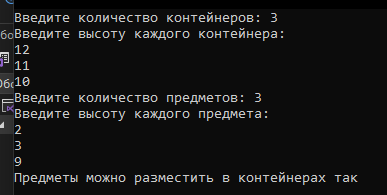
else {

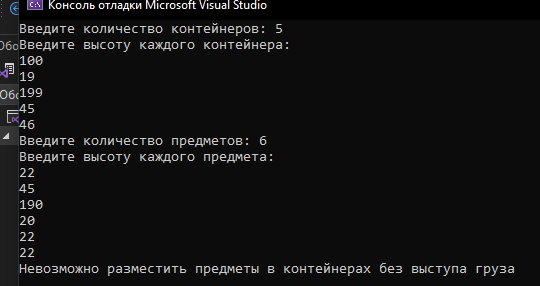
cout << "Невозможно разместить предметы в контейнерах без выступа груза\n";

}

return 0;

}





## Python:

from typing import List

def canFit(containers: List[int], items: List[int], containerIndex: int) -> bool:

if containerIndex == len(containers):

return True # All containers have been processed successfully

for i in range(len(items)):

if items[i] <= containers[containerIndex]:

itemHeight = items[i]

containers[containerIndex] -= itemHeight

if canFit(containers, items, containerIndex + 1):

return True

containers[containerIndex] += itemHeight # Revert changes

return False

def main():

N = int(input("Введите количество контейнеров: "))

containers = [int(input("Введите высоту каждого контейнера: ")) for \_ in range(N)]

M = int(input("Введите количество предметов: "))

items = [int(input("Введите высоту каждого предмета: ")) for \_ in range(M)]

if canFit(containers, items, 0):

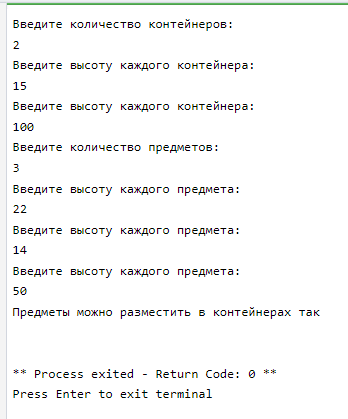
print("Предметы можно разместить в контейнерах так")

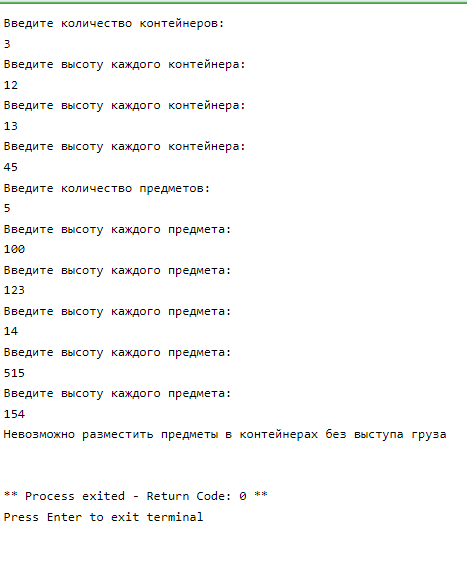
else:

print("Невозможно разместить предметы в контейнерах без выступа груза")

if \_\_name\_\_ == "\_\_main\_\_":

main()





# Задание 2. Дана закодированная строка, верните её декодированную версию. Правило кодирования: k[encoded\_string], где encoded\_string квадратные скобки повторяются ровно k раз. Обратите внимание, что k это гарантированно положительное целое число. Вы можете предположить, что входная строка всегда действительна; нет лишних пробелов, правильные квадратные скобки и т.д. Кроме того, вы можете предположить, что исходные данные не содержат никаких цифр и что цифры предназначены только для тех повторяющихся чисел k. Например, не будет ввода типа 3a или 2[4].

# С++:

## #include <iostream>

## #include <string>

## using namespace std;

## string decodeString(const string& s, size\_t& i) {

## string res = "";

## while (i < s.length() && s[i] != ']') {

## if (!isdigit(s[i])) {

## res += s[i];

## i++;

## }

## else {

## int k = 0;

## while (i < s.length() && isdigit(s[i])) {

## k = k \* 10 + (s[i] - '0');

## i++;

## }

## i++; // Пропускаем '['

## string decodedStr = decodeString(s, i);

## i++; // Пропускаем ']'

## for (int j = 0; j < k; j++) {

## res += decodedStr;

## }

## }

## }

## return res;

## }

## string decodeString(const string& s) {

## size\_t i = 0;

## return decodeString(s, i);

## }

## int main() {

## setlocale(LC\_ALL, "Rus");

## cout << "Введите: " << endl;

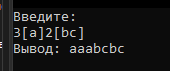
## string s;

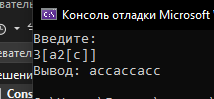
## cin >> s;

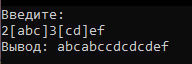
## cout << "Вывод: " << decodeString(s) << endl;

## return 0;

## }







## Python:

def decode\_string(s, i):

res = ""

while i[0] < len(s) and s[i[0]] != ']':

if not s[i[0]].isdigit():

res += s[i[0]]

i[0] += 1

else:

k = 0

while i[0] < len(s) and s[i[0]].isdigit():

k = k \* 10 + int(s[i[0]])

i[0] += 1

i[0] += 1 # Пропускаем '['

decoded\_str = decode\_string(s, i)

i[0] += 1 # Пропускаем ']'

res += decoded\_str \* k

return res

def decode\_string\_main(s):

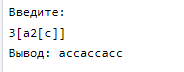
i = [0]

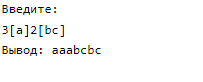
return decode\_string(s, i)

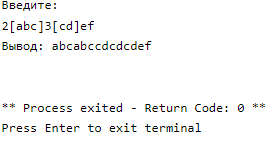
if \_\_name\_\_ == "\_\_main\_\_":

s = input("Введите: ")

print("Вывод:", decode\_string\_main(s))







**Вывод:**

В ходе выполнения лабораторной работы №7 были выполнены которые были поставлены на изучения изучение рекурсивного программирования, методов разработки эффективных алгоритмов.

**Ссылка на репозиторий:** [Нажмите](https://github.com/astnq/laba7)