**Trần Hoàng Triển - MSSV:46.01.104.196**

**Cài đặt bài toán balo 1**

**Dữ liệu vào:**

dòng 1: chứa 2 số n và w. Trong đó “n” là số lượng món đồ, “w” tổng khối lượng balo chứa

dòng 2: chứa “n” số nguyên là khối lượng từng món đồ a[0], a[1], ... ,a[n-1]

Ví dụ:

A screenshot of a computer

Description automatically generated with low confidence

**Dữ liêu ra :**

Dòng 1: khối lượng tối ưu của balo sau khi chọn các món hàng

Dòng 2: thứ tự các món hàng được chọn

A white square with black numbers

Description automatically generated with low confidence

**Miêu tả cách xử lý dữ liệu như sau:**

* Sau khi chúng ta nhập giá trị `n` (số lượng phần tử) và `w` (giới hạn trọng lượng) từ bàn phím.

Đoạn code trên thực hiện các bước sau để xử lý dữ liệu:

* Sau đó, chương trình cấp phát một mảng `a` có kích thước `n` để lưu trữ các phần tử.
* Hàm `**creArr**` được gọi để nhập giá trị cho các phần tử của mảng `a`. Đầu vào của hàm `creArr` là `n` và con trỏ `a`. Trong hàm `creArr`, chương trình sử dụng vòng lặp để lần lượt nhập giá trị của các phần tử và lưu vào mảng `a` thông qua con trỏ.
* Sau khi nhập xong, hàm `**findMax**` được gọi để tìm cặp phần tử có tổng lớn nhất không vượt quá giới hạn trọng lượng `w`. Đầu vào của hàm `findMax` là `n`, con trỏ `a` và giá trị `w`.
* Trong hàm `**findMax**`, chương trình duyệt qua tất cả các cặp phần tử trong mảng `a` bằng cách sử dụng hai vòng lặp lồng nhau. Tại mỗi cặp phần tử, chương trình tính tổng của hai phần tử và kiểm tra xem tổng đó có vượt quá giới hạn `w` hay không. Nếu vượt quá, chương trình bỏ qua cặp phần tử đó. Nếu không, chương trình so sánh tổng với giá trị `max1` (giá trị tổng lớn nhất hiện tại) và cập nhật `max1` nếu tổng lớn hơn. Chương trình cũng lưu các chỉ số `x` và `y` của cặp phần tử có tổng lớn nhất.
* Sau khi duyệt qua tất cả các cặp phần tử, chương trình in ra giá trị tổng lớn nhất `**max1**` và hai chỉ số `x` và `y` tương ứng với cặp phần tử có tổng lớn nhất.
* Với dữ liệu nhập vào `n = 3`, `w = 10` và các phần tử là `3`, `5`, `7`, kết quả của chương trình sẽ là tổng lớn nhất không vượt qua **w=10.**
* Chương trình in ra tổng lớn nhất `max1` là `10` và hai chỉ số `x` và `y` tương ứng với cặp phần tử có tổng lớn nhất, cặp phần tử có tổng lớn nhất là `3` và `7`, và các chỉ số tương ứng là `0` và `2`. Do đó, chương trình in ra:

A white square with black numbers

Description automatically generated with low confidence

* Cuối cùng, chương trình giải phóng bộ nhớ đã được cấp phát cho mảng `a` bằng câu lệnh `delete[ ] a`, đảm bảo không có rò rỉ bộ nhớ và giải phóng tài nguyên sau khi hoàn thành xử lý.

Thêm Tiêu đề (Định dạng > Kiểu đoạn). Khi đó, chúng sẽ xuất hiện trong mục lục.

**Để thực thi chương trình trên, bạn có thể làm theo các bước sau:**

1. Mở trình biên dịch C++ (như Dev-C++, Visual Studio,).

2. Tạo một tệp tin mới và sao chép đoạn mã vào tệp tin đó.

3. Lưu tệp tin với phần mở rộng `.cpp` (ví dụ: `**balo1.cpp**`).

4. Tiến hành chạy chương trình và nhập dữ liệu từ bàn phím vào gồm n, w và dãy số nguyên giá trị của món đồ.

5. Trình biên dịch sẽ thực thi chương trình và hiển thị kết quả.

**Source code:**

#include <iostream>

using namespace std;

void creArr(int n, int \*a);

void findMax(int n, int \*a, int w);

int main()

{

int n, w;

cin >>n>> w;

int \*a=new int[n];

creArr(n,a);

findMax(n,a,w);

delete[] a;

return 0;

}

void creArr(int n, int \*a){

for(int i=0;i<n;++i)

cin>>\*(a+i);

}

void findMax(int n, int \*a, int w){

int max1=0, sum, x, y;

for(int i=0;i<n-1;++i)

for(int j=i+1;j<n;++j){

sum=\*(a+i) + \*(a+j);

if(sum>w) continue;

if(sum<max1) continue;

max1=sum;

x=i;

y=j;

}

cout<<max1<<endl<<x<<" "<<y;

}