**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**



**MÔN CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**

**BÀI TẬP THỰC HÀNH 2**

GVHD: Nguyễn Đình Hiển

🙡🙢 Tp. Hồ Chí Minh, 03/2024 🙠🙣**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*……., ngày……...tháng……năm 2024*

**Người nhận xét**

*(Ký tên và ghi rõ họ tên****)***

**MỤC LỤC**

**MỤC LỤC**

[Bài 1: Thiết kế giải thuật dạng vét cạn để tìm tập con có tích lớn nhất của dãy a, cài đặt chương trình bằng C/C++/Python. 6](#_Toc160658747)

[Bài 2: Tìm hiểu giải thuật MergeSort, hãy cho biết đây là dạng nào trong các loại: vét cạn/chia để trị/tham lam/qui hoạch động. 7](#_Toc160658748)

[Bài 3: Bài toán đổi tiền: Có M loại tiền mệnh giá S1, S2, …, SM; số lượng mỗi loại không hạn chế. Cần xác định số cách đổi số tiền N đồng thành các tờ tiền trong M loại đã cho. 7](#_Toc160658749)

**DANH MỤC BẢNG**

[Bảng 1 Chương trình bài tập 1 6](#_Toc160663323)

[Bảng 2 Chương trình bài tập 3 8](#_Toc160663324)

**DANH MỤC HÌNH ẢNH**

[Hình ảnh 1 Kết quả chạy lần 1 bài 1 7](#_Toc160663055)

[Hình ảnh 2 Kết quả chạy lần 2 bài 1 7](#_Toc160663056)

[Hình ảnh 3 Kết quả chạy lần 3 bài 1 8](#_Toc160663057)

[Hình ảnh 4 Kết quả chạy lần 4 bài 1 8](#_Toc160663058)

[Hình ảnh 5 Kết quả chạy lần 5 bài 1 8](#_Toc160663059)

[Hình ảnh 6 Kết quả chạy lần 1 bài 3 10](#_Toc160663060)

[Hình ảnh 7 Kết quả chạy lần 2 bài 3 10](#_Toc160663061)

[Hình ảnh 8 Kết quả chạy lần 3 bài 3 10](#_Toc160663062)

[Hình ảnh 9 Kết quả chạy lần 4 bài 3 10](#_Toc160663063)

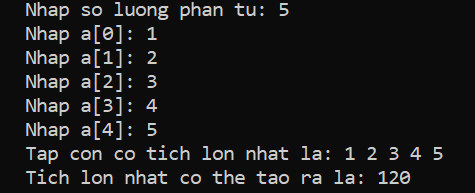
[Hình ảnh 10 Kết quả chạy lần 5 bài 3 10](#_Toc160663064)

**NỘI DUNG BÀI LÀM**

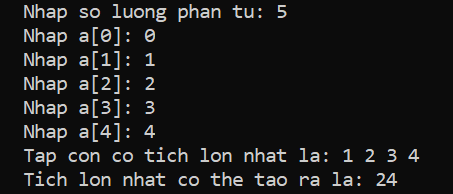
# Bài 1: Thiết kế giải thuật dạng vét cạn để tìm tập con có tích lớn nhất của dãy a, cài đặt chương trình bằng C/C++/Python.

Bảng 1 Chương trình bài tập 1

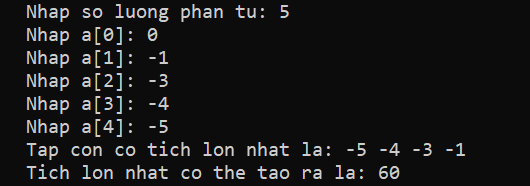
|  |
| --- |
| #include <iostream>  #include <algorithm>  #include <vector>  using namespace std;  int TichLN(vector<int>& a) {  sort(a.begin(), a.end());  int num = 0;    a.erase(std::remove(a.begin(), a.end(), num), a.end());  int n = a.size();  if (a[0] > 0) {  int kq = 1;  cout << "Tap con co tich lon nhat la: ";  for (int i = 0; i < n; i++) {  kq \*= a[i];  cout << a[i] << " ";  }  return kq;  }  else {  int dem = 0; int kq = 1;  while (dem < n && a[dem] < 0) dem++;  if (dem % 2 != 0) {  dem--;  cout << "Tap con co tich lon nhat la: ";  for (int i = 0; i < dem; i++) {  kq \*= a[i];  cout << a[i] << " ";  }  for (int i = dem + 1; i < n; i++) {  kq \*= a[i];  cout << a[i] << " ";  }  return kq;  }  else  {  cout << "Tap con co tich lon nhat la: ";  for (int i = 0; i < n; i++) {  kq \*= a[i];  cout << a[i] << " ";  }  return kq;  }  }  }  int main() {  int n;  cout << "Nhap so luong phan tu: "; cin >> n;  vector<int> a(n);  for (int i = 0; i < n; ++i) {  cout << "Nhap a[" << i << "]: "; cin >> a[i];  }  int result = TichLN(a);  cout << "\nTich lon nhat co the tao ra la: " << result;  return 0;  } |



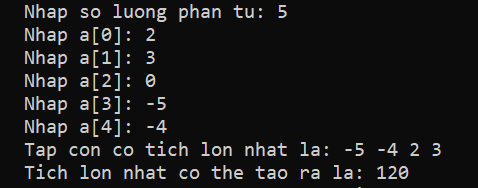
Hình ảnh 1 Kết quả chạy lần 1 bài 1



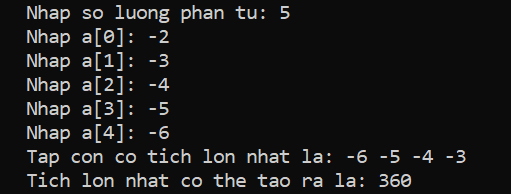
Hình ảnh 2 Kết quả chạy lần 2 bài 1



Hình ảnh 3 Kết quả chạy lần 3 bài 1



Hình ảnh 4 Kết quả chạy lần 4 bài 1



Hình ảnh 5 Kết quả chạy lần 5 bài 1

# Bài 2: Tìm hiểu giải thuật MergeSort, hãy cho biết đây là dạng nào trong các loại: vét cạn/chia để trị/tham lam/qui hoạch động.

* MergeSort là một giải thuật chia để trị (divide and conquer). Giải thuật này chia mảng đầu vào thành hai nửa, sắp xếp từng nửa riêng lẻ, sau đó kết hợp hai nửa đã sắp xếp để tạo ra dãy hoàn chỉnh đã được sắp xếp.

# Bài 3: Bài toán đổi tiền: Có M loại tiền mệnh giá S1, S2, …, SM; số lượng mỗi loại không hạn chế. Cần xác định số cách đổi số tiền N đồng thành các tờ tiền trong M loại đã cho.

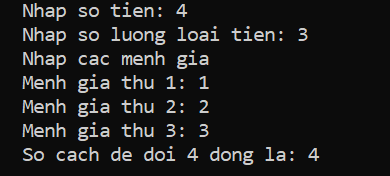
**Ví dụ: N=4, M=3 và S = {1, 2, 3}. Có 4 cách đổi tiền: 4 tờ 1; 2 tờ 1-1 tờ 2; hai tờ 2; 1 tờ 1 - 1 tờ 3.**

**Hãy lựa chọn dạng giải thuật thích hợp để giải quyết bài toán. Giải thích lý do chọn.**

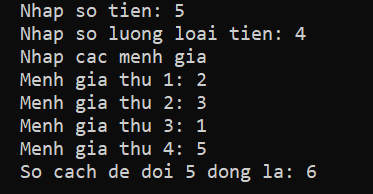
* Bài toán được giải bằng phương pháp Quy hoạch động (dynamic programming) vì bài toán đã cho có tính chất trùng lặp (Có nhiều cách đổi tiền giống nhau) nên sử dụng quy hoạch động sẽ tối ưu hơn vì giúp tránh tính toán lại các bài toán con đã được giải quyết, giúp giảm thời gian tính toán và không gian lưu trữ so với các phương pháp khác.

Bảng 2 Chương trình bài tập 3

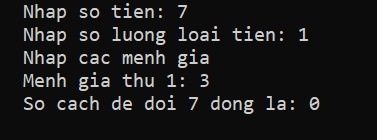
|  |
| --- |
| #include <iostream>  #include <vector>  #include <algorithm>  using namespace std;  int N, M;  vector<int> MenhGia;  void Nhap() {  cout << "Nhap so tien: ";  cin >> N;  cout << "Nhap so luong loai tien: ";  cin >> M;  MenhGia.resize(M);  cout << "Nhap cac menh gia\n";  for (int i = 0; i < M; ++i) {  cout << "Menh gia thu " << i + 1 << ": ";  cin >> MenhGia[i];  }  std::sort(MenhGia.begin(), MenhGia.end());  }  void Change\_Money(int N, vector<int> MenhGia) {  vector<int> dem(N + 1, 0);  dem[0] = 1;  for (int i = 0; i < M; ++i) {  for (int j = MenhGia[i]; j <= N; ++j) {  dem[j] += dem[j - MenhGia[i]];  }  }  cout << "So cach de doi " << N << " dong la: " << dem[N] << endl;  }  int main() {  Nhap();  Change\_Money(N, MenhGia);  return 0; |

****

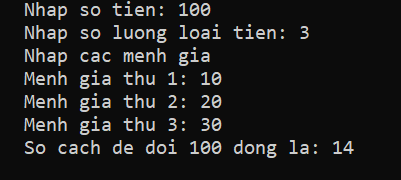
Hình ảnh 6 Kết quả chạy lần 1 bài 3

****

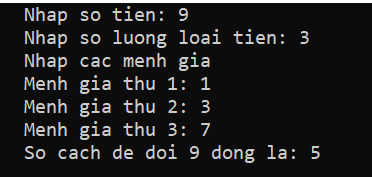
Hình ảnh 7 Kết quả chạy lần 2 bài 3

****

Hình ảnh 8 Kết quả chạy lần 3 bài 3

****

Hình ảnh 9 Kết quả chạy lần 4 bài 3

****

Hình ảnh 10 Kết quả chạy lần 5 bài 3