Mục tiêu:

1. ***Diễn đạt và hiện thực tác vụ hướng đệ quy.***
2. ***Đệ quy với cấu trúc dữ liệu là danh sách đặc***
3. ***So sánh thời gian thực hiện các tác vụ trên danh sách đặc(array) với 2 kỹ thuật duyệt tuần tự và đệ quy.***

Vấn đề 4: Lãi suất kép

Ngân hàng A có lãi suất cố định là x%. Đem khoản tiền y gửi vào ngân hàng A trong khoảng thời gian n năm. Tính số tiền thu được (vốn + lãi).

Yêu cầu:

1. Gọi P(n) là số tiền thu được sau n năm gửi số tiền y vào A. Lập công thức đệ quy tính P(n)
2. Cài đặt các hàm đệ quy ***float lai_kep(float y, int n);*** *trả về số tiền thu được sau n năm*
3. Cài đặt chương trình hiện thực vấn đề 4 bằng hàm int main(); để kiểm tra tính đúng bằng thực nghiệm.
    1. Nhập dữ liệu hợp lệ cho n và y
    2. Hiển thị giá trị của P(n) trên màn hình tương ứng với dữ liệu vào n và y từ bản phím

Test:

Input	Output
n = 0 y = 0	
n = 1 y =10	
n = 10^4 y =10	
n = 10^6 y =10	

Vấn đề 5 : Đệ quy trên mảng

Dãy Q gồm n số nguyên không có thứ tự. Q được lưu trữ ở bộ nhớ trong bằng mảng int Q[n].

Yêu cầu:

4. Cài đặt hàm ***void input(int Q[], int &n)*** để sinh ngẫu nhiên n phần tử ($1 \le n \le 10^6$) và lưu vào mãng Q[].
5. Cài đặt hàm ***void output(int Q[], int n)*** để hiển thị các phần tử của Q[] lên màn hình.
6. Dưới đây là gợi ý xây dựng định nghĩa đệ quy để tính tổng n phần tử của dãy Q.

Sum(q,n) = q[0] + q[1] + q[2] + ... + q[n-2] +q[n-1]
Sum(q,n-1)

$$Sum(q,n) = \begin{cases} 0 \text{ nếu } n = 0 \\ q[n-1] + Sum(q,n-1) \text{ nếu } n > 0 \end{cases}$$

7. Cài đặt hàm ***long sum_rec(int Q[], int n)*** trả về tổng các phần tử của dãy số Q bằng kỹ thuật đệ quy được mô tả ở 6.
8. Dưới đây là gợi ý để xây dựng định nghĩa đệ quy tìm phần tử lớn nhất của mảng Q.

- Điều kiện biên: Mảng 1 phần tử thì trị lớn nhất là q[0].
- Giải thuật chung:
  - Max(q,n) = q[0] , q[1] , q[2] , ... , q[n-2] , q[n-1]
    - **Max(q,n-1)**
  - $\boldsymbol{Max(q,n)} = \begin{cases} \boldsymbol{q[0]\ nếu\ n = 1} \\ \boldsymbol{q[n-1] > Max(q,n-1)?\ q[n-1]: Max(q,n-1)} \end{cases}$

9. Cài đặt hàm ***int max_rec(int Q[], int n)*** trả về giá trị ***lớn nhất*** của dãy số Q bằng kỹ thuật đệ quy được mô tả ở 8.
10. Định nghĩa đệ quy cho tác vụ tìm phần tử có giá trị nhỏ nhất trong dãy số Q. Từ đó cài đặt hàm ***int min_rec(int Q[], int n)*** trả về giá trị ***bé nhất*** của dãy số Q bằng kỹ thuật đệ quy.
11. Cài đặt hàm ***long sum(int Q[], int n)*** trả về tổng các phần tử của dãy số Q bằng phương pháp duyệt tuần tự trên mảng Q.
12. Cài đặt chương trình hiện thực ***vấn đề 5*** bằng hàm ***int main();***
    - Tạo dãy số Q gồm n số bằng cách sinh ngẫu nhiên (1≤n≤100)
    - Hiển thị các phần tử của dãy Q
    - Hiển thị phần tử lớn nhất và nhỏ nhất của dãy Q.
    - Tạo dãy số Q gồm n phần tử bằng cách sinh ngẫu nhiên ($10^4 \le n \le 10^9$). Thực hiện tính tổng của Q theo hai cách Đệ quy và Không đệ quy và so sánh thời gian thực hiện của mỗi giải thuật với cùng bộ dữ liệu Q(n). Lập báo cáo theo mẫu sau:

	n=10^4	**n=10^5**	**n=10^6**	**n=10^7**	**n=10^9**
sum	…(s)	…(s)	…(s)	…(s)	…(s)
sum_rec	…(s)	…(s)	…(s)	…(s)	…(s)

Đoạn chương trình hướng dẫn kỹ thuật đo thời gian:

```
#include<iostream>
#include<time.h>
#include <stdlib.h>
using namespace std;
long sum_rec(int Q[], int n) ;
long sum(int Q[], int n);
int main()
{       int n = 10000;//10^4, 10^5, 10^6, ...
 clock_t start= clock();       sum(n);       clock_t end = clock();  //Đo thời gian không đq
 cout<<"Time = "<<(float)(end-start)/100<<"(s)"<<endl;
 clock_t start_dq= clock();  sum_rec(n);   clock_t end_dq = clock(); // Đo thời gian đệ quy
 cout<<"Time = "<<(float)(end_dq-start_dq)/100<<"(s)"<<endl;
}
```