**Bài 1: Dãy số ACSLacci**

|  |
| --- |
| #include <iostream>  #include <vector>  using namespace std;  int main()  {  freopen("ACSLacci.INP", "r", stdin);  freopen("ACSLacci.OUT", "w", stdout);  int x1, x2, x3, n;  cin >> x1 >> x2 >> x3 >> n;  if (n <= 0) {  cout << "n > 0!" << endl;  return 1;  }  if (n == 1) {  cout << x1 << endl;  return 0;  }  else if (n == 2) {  cout << x2 << endl;  return 0;  }  else if (n == 3) {  cout << x3 << endl;  return 0;  }  vector<int> a = {x1, x2, x3};  for (int i = 3; i < n; ++i) {  int next = a[i-1] + a[i-2] + a[i-3] - 3;  a.push\_back(next);  }  cout << a[n-1] << endl;  return 0;  } |

**Bài 2: Đường đi đến số 0**

|  |
| --- |
| #include <iostream>  using namespace std;  int main()  {  freopen("ZEROPATH.INP", "r", stdin);  freopen("ZEROPATH.OUT", "w", stdout);  int n, result[10000] = {0}, count = 1;  cin >> n;  for (int i = 0, num; i < count; ++i) {  if (i == 0) {  num = n;  }  else {  num = result[i];  }  for (int x = 1; x <= num / x; ++x) {  if (num % x == 0) {  int newValue = (x - 1) \* (num / x + 1);  if (newValue >= 0) {  bool exists = false;  for (int j = 0; j < count; ++j) {  if (result[j] == newValue) {  exists = true;  }  }  if (!exists) {  result[count++] = newValue;  }  }  }  }  }    for (int i = 1; i < count; ++i) {  for (int j = i; j > 0 && result[j] < result[j - 1]; --j) {  swap(result[j], result[j - 1]);  }  }  cout << count << endl;  for (int i = 0; i < count; ++i) {  cout << result[i] << endl;  }  return 0;  } |

**Bài 3: Sóng nước**

|  |
| --- |
| #include <iostream>  #include <string>  using namespace std;  bool isWave(const string& numStr, int length)  {  if (length <= 1) {  return true;  }  for (int i = 1; i < length; ++i) {  if ((numStr[i - 1] < numStr[i]) == (i % 2 == 0)) {  return false;  }  }  return true;  }  int main()  {  freopen("WA.INP", "r", stdin);  freopen("WA.OUT", "w", stdout);  string numStr;  cin >> numStr;  int maxWave = 1;  int length = numStr.length();  for (int i = 1; i <= length; ++i) {  if (isWave(numStr, i)) {  maxWave = i;  }  }  cout << maxWave << endl;  return 0;  } |