|  |
| --- |
| TRƯỜNG ĐẠI HỌC BÁCH KHOA  **KHOA CÔNG NGHỆ THÔNG TIN** |

**ĐỀ THI VÀ BÀI LÀM**

Tên học phần: Toán ứng dụng CNTT

Mã học phần: Hình thức thi: *Tự luận có giám sát*

Đề số: **Đ0003** Thời gian làm bài: 90 phút *(không kể thời gian chép/phát đề)*

**Họ tên: Đinh Minh Tuệ Lớp**:23T\_DT1 **MSSV**: 102230223

Sinh viên làm bài trực tiếp trên tệp này, lưu tệp với định dạng MSSV\_HọTên.pdf và nộp bài thông qua MS Teams.

***Câu 1*** (*2.0 điểm*): Viết chương trình (có sử dụng hàm) thực hiện công việc sau, biết rằng n=100:

, .

a) *(1.0 điểm)* Kiểm tra F(n) có phải số nguyên tố hay không?

|  |
| --- |
| **# Trả lời: Dán code bên dưới:**    **# Trả lời: Dán kết quả thực thi vào bên dưới:**  **#include <stdio.h>**  **#include <math.h>**  **unsigned long long F(int n)**  **{**  **if (n == 1 || n == 2) return 1;**  **unsigned long long f1 = 1, f2 = 1, fn;**  **for (int i = 3; i <= n; i++)**  **{**  **fn = f1 + f2;**  **f1 = f2;**  **f2 = fn;**  **}**  **return f2;**  **}**  **int isPrime(unsigned long long x)**  **{**  **if (x < 2) return 0;**  **if (x == 2) return 1;**  **if (x % 2 == 0) return 0;**  **for (unsigned long long i = 3; i \* i <= x; i += 2)**  **if (x % i == 0)**  **return 0;**  **return 1;**  **}**  **int main()**  **{**  **int n = 100;**  **unsigned long long fn = F(n);**  **printf("F(%d) = %llu\n", n, fn);**  **if (isPrime(fn))**  **printf("F(%d) la so nguyen to\n", n);**  **else**  **printf("F(%d) khong la so nguyen to\n", n);**  **return 0;**  **}** |

1. *(1.0 điểm)* Trong n số Fibonacci đầu tiên, hãy liệt kê các số nào là số nguyên tố và đếm tổng số đó.

|  |
| --- |
| **# Trả lời: Dán code vào bên dưới:**  **#include <stdio.h>**  **#include <math.h>**  **int F(int n) {**  **if (n == 1 || n == 2) return 1;**  **return F(n - 1) + F(n - 2);**  **}**  **int isPrime(int x) {**  **if (x < 2) return 0;**  **if (x == 2) return 1;**  **if (x % 2 == 0) return 0;**  **for (int i = 3; i <= sqrt(x); i += 2)**  **if (x % i == 0) return 0;**  **return 1;**  **}**  **int main() {**  **int n;**  **printf("Nhap n: ");**  **scanf("%d", &n);**  **int count = 0;**  **printf("Cac so Fibonacci la so nguyen to:\n");**  **for (int i = 1; i <= n; i++) {**  **int fib = F(i);**  **if (isPrime(fib)) {**  **printf("F(%d) = %d\n", i, fib);**  **count++;**  **}**  **}**  **printf("Tong so: %d\n", count);**  **return 0;**  **}**  **# Trả lời: Dán kết quả thực thi vào bên dưới:** |

***Câu 2*** (*1.5 điểm*): Cho ma trận A. Viết chương trình (có sử dụng hàm) thực hiện phân rã ma trận A bằng phương pháp SVD.

|  |
| --- |
| **# Trả lời:** **Dán code vào bên dưới**  **#include <cstdio>**  **#include <vector>**  **#include <cmath>**  **#include <algorithm>**  **#include <Eigen/Dense>**  **using namespace std;**  **void print(const char\* name, const vector<vector<double>>& M)**  **{**  **printf("%s =\n", name);**  **for (auto& r : M) {**  **printf("[ ");**  **for (double x : r)**  **printf("% .6f ", x);**  **printf("]\n");**  **}**  **printf("\n");**  **}**  **vector<vector<double>> transpose(const vector<vector<double>>& A)**  **{**  **int m = A.size(), n = A[0].size();**  **vector<vector<double>> T(n, vector<double>(m));**  **for(int i=0;i<m;i++)**  **for(int j=0;j<n;j++)**  **T[j][i] = A[i][j];**  **return T;**  **}**  **vector<vector<double>> multiply(**  **const vector<vector<double>>& A,**  **const vector<vector<double>>& B)**  **{**  **int m = A.size(), n = B[0].size(), p = B.size();**  **vector<vector<double>> C(m, vector<double>(n, 0));**  **for(int i=0;i<m;i++)**  **for(int j=0;j<n;j++)**  **for(int k=0;k<p;k++)**  **C[i][j] += A[i][k]\*B[k][j];**  **return C;**  **}**  **vector<vector<double>> ATA(const vector<vector<double>>& A)**  **{**  **return multiply(transpose(A), A);**  **}**  **int main()**  **{**  **int m, n;**  **printf("Nhap m n: ");**  **scanf("%d%d", &m, &n);**  **vector<vector<double>> A(m, vector<double>(n));**  **for(int i=0;i<m;i++)**  **for(int j=0;j<n;j++)**  **scanf("%lf", &A[i][j]);**  **print("A", A);**  **vector<vector<double>> AtA = ATA(A);**  **Eigen::MatrixXd M(n,n);**  **for(int i=0;i<n;i++)**  **for(int j=0;j<n;j++)**  **M(i,j) = AtA[i][j];**  **Eigen::SelfAdjointEigenSolver<Eigen::MatrixXd> es(M);**  **vector<double> sigma(n);**  **vector<vector<double>> V(n, vector<double>(n));**  **for(int i=0;i<n;i++)**  **{**  **sigma[i] = sqrt(max(0.0, es.eigenvalues()(n-1-i)));**  **for(int j=0;j<n;j++)**  **V[j][i] = es.eigenvectors()(j, n-1-i);**  **}**  **vector<vector<double>> U(m, vector<double>(n, 0));**  **for(int i=0;i<n;i++)**  **if(sigma[i] > 1e-12)**  **for(int r=0;r<m;r++)**  **for(int c=0;c<n;c++)**  **U[r][i] += A[r][c]\*V[c][i]/sigma[i];**  **vector<vector<double>> S(n, vector<double>(n,0));**  **for(int i=0;i<n;i++) S[i][i] = sigma[i];**  **print("U", U);**  **print("Sigma", S);**  **print("V^T", transpose(V));**  **print("U\*Sigma\*V^T", multiply(multiply(U,S), transpose(V)));**  **return 0;**  **}**  **# Trả lời:** **Dán kết quả thực thi** vào bên dưới biết rằng , sai số . |

***Câu 3*** (*3.0 điểm*): Cho 18 điểm trong không gian Oxy như sau: *(1,0), (4,0), (6,2), (5,5), (3,6), (0,3), (2,0), (0,2), (3,3), (4,1), (2,2),(4,4), (2,4), (1,2),(1,5),(3,1),(3,5),(5,3).*

1. *(0.5 điểm) Mô tả thuật toán xác định bao lồi của tập điểm trên (dạng sơ đồ khối hoặc mã giả).*

|  |
| --- |
| **# Trả lời:** **dán sơ đồ khối hoặc mã giả:**  - Sắp xếp các điểm trong P: sắp xếp tăng đần theo x, nếu x bằng y thì sắp xếp theo y      if (p1->x == p2->x)          return (p1->y > p2->y) - (p1->y < p2->y);      return (p1->x > p2->x) - (p1->x < p2->x);   * Khởi tạo danh sách H rỗng để tạo bao lồi dưới * For mỗi điểm Pi trong P theo thứ tự đã sắp xếp: * While H có ít nhất 2 điểm và 3 điểm (H[k-2], H[k-1], Pi) không tạo thành rẽ trái: loại bỏ điểm H[k-1] khỏi H, thêm Pi vào H   - Lưu vị trí bắt đầu của bao lồi trên là t = số phần tử hiện tại của H + 1  - For mỗi điểm Pi trong P theo thứ tự ngược lại (từ phải sang trái)  - While H có ít nhất t điểm và 3 điểm (H[k-2], H[k-1], Pi) không tạo thành rẽ trái:loại bỏ điểm H[k-1] khỏi H và thêm Pi vào H  - Loại bỏ điểm cuối cùng vì trùng với điểm đầu, trả về danh sách H là bao lồi của tập điểm |

1. *(1.0 điểm)* Viết chương trình tìm bao lồi, sau đó tính cạnh nhỏ nhất của đa giác lồi tìm được.

|  |
| --- |
| **# Trả lời:** **Dán code bên dưới:**  #include <stdio.h>  #include <stdlib.h>  #include <math.h>  typedef struct {      double x, y;  } Point;  int cmp(const void \*a, const void \*b) {      Point \*p1 = (Point \*)a;      Point \*p2 = (Point \*)b;      if (p1->x == p2->x)          return (p1->y > p2->y) - (p1->y < p2->y);      return (p1->x > p2->x) - (p1->x < p2->x);  }  double cross(Point O, Point A, Point B) {      return (A.x - O.x)\*(B.y - O.y) - (A.y - O.y)\*(B.x - O.x);  }  double dist(Point A, Point B) {      return sqrt((A.x-B.x)\*(A.x-B.x) + (A.y-B.y)\*(A.y-B.y));  }  //tim bao loi  int convexHull(Point \*P, int n, Point \*H) {      qsort(P, n, sizeof(Point), cmp);      int k = 0;      for (int i = 0; i < n; i++) {          while (k >= 2 && cross(H[k-2], H[k-1], P[i]) <= 0) k--;          H[k++] = P[i];      }      int t = k + 1;      for (int i = n-2; i >= 0; i--) {          while (k >= t && cross(H[k-2], H[k-1], P[i]) <= 0) k--;          H[k++] = P[i];      }      return k-1;  }  int main() {      Point P[] = {          {1,0},{4,0},{6,2},{5,5},{3,6},{0,3},          {2,0},{0,2},{3,3},{4,1},{2,2},{4,4},          {2,4},{1,2},{1,5},{3,1},{3,5},{5,3}      };      int n = 18;      Point H[50];      int m = convexHull(P, n, H);      printf("Cac dinh bao loi:\n");      for (int i = 0; i < m; i++)          printf("(%.0f, %.0f)\n", H[i].x, H[i].y);      double minEdge = 1e9;      for (int i = 0; i < m; i++) {          double d = dist(H[i], H[(i+1)%m]);          if (d < minEdge) minEdge = d;      }   printf("\nCanh nho nhat cua bao loi = %.4f\n", minEdge);   return 0;  }  **# Trả lời:** **Dán kết quả thực thi vào bên dưới:** |

c)  *(1.5 điểm)* Xác định diện tích của đa giác bao lồi vừa tìm được. Xác định số lượng các điểm nằm bên trong bao lồi và liệt kê chúng.

|  |
| --- |
| **# Trả lời:** **Dán code bên dưới:**  #include <stdio.h>  #include <stdlib.h>  #include <math.h>  typedef struct {      double x, y;  } Point;  int cmp(const void \*a, const void \*b) {      Point \*p1 = (Point \*)a;      Point \*p2 = (Point \*)b;      if (p1->x == p2->x)          return (p1->y > p2->y) - (p1->y < p2->y);      return (p1->x > p2->x) - (p1->x < p2->x);  }  double cross(Point O, Point A, Point B) {      return (A.x - O.x)\*(B.y - O.y) - (A.y - O.y)\*(B.x - O.x);  }  double dist(Point A, Point B) {      return sqrt((A.x-B.x)\*(A.x-B.x) + (A.y-B.y)\*(A.y-B.y));  }  double polygonArea(Point \*H, int m) {      double area = 0;      for (int i = 0; i < m; i++) {          int j = (i+1)%m;          area += H[i].x\*H[j].y - H[j].x\*H[i].y;      }      return fabs(area)/2;  }  int isInsideConvex(Point \*H, int m, Point P) {      for (int i = 0; i < m; i++) {          Point A = H[i];          Point B = H[(i+1)%m];          if (cross(A, B, P) <= 0) return 0;      }      return 1;  }  int main() {      Point P[] = {          {1,0},{4,0},{6,2},{5,5},{3,6},{0,3},          {2,0},{0,2},{3,3},{4,1},{2,2},{4,4},          {2,4},{1,2},{1,5},{3,1},{3,5},{5,3}      };      int n = 18;      Point H[50];        double area = polygonArea(H, m);      printf("\nDien tich bao loi = %.4f\n", area);      int count = 0;      printf("\nCac diem nam ben trong bao loi:\n");      for (int i = 0; i < n; i++) {          if (isInsideConvex(H, m, P[i])) {              printf("(%.0f, %.0f)\n", P[i].x, P[i].y);              count++;          }      }      printf("\nSo diem nam ben trong bao loi = %d\n", count);      return 0;  }  **# Trả lời:** **Dán kết quả thực thi vào bên dưới:** |

***Câu 4*** (*2.0 điểm*): Cho hàm số

1. *(0.5 điểm) Trình bày thuật toán tối ưu hàm số đã cho* sử dụng phương pháp *gradient descent với momentum*, biết rằng tham số học (learning rate) , hệ số động lượng là .

|  |
| --- |
| **# Trả lời: dán sơ đồ khối hoặc mã giả:**  Lặp Gradient Descent với Momentum  For i = 1 đến N do  Tính gradient:  gx ← ∂f/∂x tại (x, y)  gy ← ∂f/∂y tại (x, y)  Cập nhật động lượng:  vx ← α \* vx + γ \* gx  vy ← α \* vy + γ \* gy  Cập nhật nghiệm mới:  x\_new ← x - vx  y\_new ← y - vy  Nếu |x\_new - x| < ε và |y\_new - y| < ε thì  Dừng thuật toán  Gán:  x ← x\_new  y ← y\_new  Kết thúc For  Trả về (x, y) là điểm cực tiểu gần đúng  Giá trị nhỏ nhất f(x, y) |

1. *(1.5 điểm)* Viết chương trình (có dùng hàm) tính giá trị bé nhất của f(x,y) sử dụng phương pháp *gradient descent với momentum* với số bước lặp *N* và sai số .

|  |
| --- |
| **# Trả lời**: **Dán code vào bên dưới:**  **#include <stdio.h>**  **#include <math.h>**  **//ham fx**  **double f(double x, double y)**  **{**  **return pow(x\*x + y - 11, 2)**  **+ pow(x + y\*y - 7, 2);**  **}**  **void grad(double x, double y, double \*gx, double \*gy)**  **{**  **\*gx = 4\*x\*(x\*x + y - 11) + 2\*(x + y\*y - 7);**  **\*gy = 2\*(x\*x + y - 11) + 4\*y\*(x + y\*y - 7);**  **}**  **void gradientDescentMomentum(**  **double x0, double y0,**  **double alpha, double gamma,**  **int N, double eps)**  **{**  **double x = x0, y = y0;**  **double vx = 0.0, vy = 0.0;   // momentum**  **double gx, gy;**  **for(int i = 1; i <= N; i++)**  **{**  **grad(x, y, &gx, &gy);**  **vx = gamma \* vx + alpha \* gx;**  **vy = gamma \* vy + alpha \* gy;**  **double x\_new = x - vx;**  **double y\_new = y - vy;**  **if (fabs(x\_new - x) < eps && fabs(y\_new - y) < eps)**  **break;**  **x = x\_new;**  **y = y\_new;**  **if (i % 100 == 0)**  **printf("Lap %d: x = %.6f, y = %.6f, f = %.6f\n",**  **i, x, y, f(x,y));**  **}**  **printf("\nKet qua\n");**  **printf("x = %.8f\n", x);**  **printf("y = %.8f\n", y);**  **printf("f(x,y) = %.10f\n", f(x,y));**  **}**  **int main()**  **{**  **double x0 = 0.0;**  **double y0 = 0.0;**  **double alpha = 0.001;   // learning rate**  **double gamma = 0.9;     // momentum**  **int N = 2000;**  **double eps = 1e-6;**  **printf("Diem khoi tao: (%.2f, %.2f)\n", x0, y0);**  **printf("Alpha = %.4f, Gamma = %.2f\n", alpha, gamma);**  **printf("So buoc lap N = %d, Sai so eps = %.1e\n\n", N, eps);**  **gradientDescentMomentum(x0, y0, alpha, gamma, N, eps);**  **return 0;**  **}**  **# Trả lời**: **Dán kết quả thực thi** với điểm khởi , tham số học học (*learning rate*) , hệ số động lượng (*momentum coefficient*) là , số bước lặp và sai số : |

***Câu 5*** (*1.5 điểm*): Một hệ thống có chế độ làm việc ở mỗi giai đoạn vận hành chỉ với các trạng thái từ 1 đến 5. Chế độ làm việc của hệ thống này được mô tả bằng ma trận chuyển như sau:

a) (*0.5 điểm*) Vẽ đồ thị biễu diễn chuỗi Markov tương ứng đã cho

|  |
| --- |
| **# Trả lời:** **Dán kết quả vào bên dưới** |

b) (*1.0 điểm*) Giả sử rằng hệ thống bắt đầu học ở trạng thái 2. Tính xác xuất hệ thống làm việc ở trạng thái 4 *sau 3 bước thời gian vận hành,* vàtrạng thái 2 *sau 5 bước thời gian vận hành*

|  |
| --- |
| **# Trả lời**: **Dán kết quả tính toán vào bên dưới:** |

Đà Nẵng, ngày 4 tháng 12 năm 2025

|  |  |
| --- | --- |
| **GIẢNG VIÊN BIÊN SOẠN ĐỀ THI** | **KHOA CÔNG NGHỆ THÔNG TIN**  **( đã duyệt)** |
|  |  |