|  |  |  |
| --- | --- | --- |
| //smallest notes  int minNotes(int bankNote){  int note =0;  while(bankNote>0){  if(bankNote>=1000){ note++;  bankNote = bankNote-1000;  }  else if(bankNote>=500){ note++;  bankNote = bankNote-500;  }  else if(bankNote>=100){ note++;  bankNote = bankNote-100;  }  else if(bankNote>=50){ note++;  bankNote = bankNote-50;  }  else if(bankNote>=20){ note++;  bankNote = bankNote-20;  }  else if(bankNote>=10){ note++;  bankNote = bankNote-10;  }  else if(bankNote>=5){ note++;  bankNote = bankNote-5;  }  else if(bankNote>=2){ note++;  bankNote = bankNote-2;  }  else{note++;  bankNote--;  }  }  return note;  }  int main(){  int taka;  cout<<"Enter the amount of TK: ";  cin>>taka;  cout<<"Min Notes:"<<minNotes(taka)<<  endl;  return 0;  } | //sorted array//for last max=(size-1)  void findMedian(int arr[],int size) {  for (int i = 0; i < size; ++i) {  for (int j = 0; j < size - 1; ++j) {  if (arr[j] > arr[j + 1]) {  int a = arr[j + 1];  arr[j + 1] = arr[j];  arr[j] = a;  }  }  }  cout << "Sorted array is: ";  for (int i = 0; i < size; ++i) {  cout << arr[i] << " ";  }  cout << endl;  }  ……………………………………………………………………………………………………  //Switching btwn 2 pairs of string  void switchPairs(string arr[], int length){  for(int i=0; i<length-1; i = i+2){  string s = arr[i];  arr[i] = arr[i+1];  arr[i+1] = s;  }  }  int main(){  string arr[7]=  {"a","bb","c","ddd","ee","f","g"};  cout<<"Before change: "<<endl;  for(int i=0; i<7; i++){  cout<<arr[i]<<" ";  }  cout<<endl;  switchPairs(arr, 7);  cout<<"After change: "<<endl;  for(int i=0; i<7; i++){  cout<<arr[i]<<" ";  }  cout<<endl;  return 0;  } | //if one word contains the same letter as the other  #include<string>  bool isanagram(string c, string d) {  int lenc = c.length();  int lend = d.length();  int counter = 0;  if (lenc == lend){  for (int i = 0; i <lenc; ++i) {  for (int j = 0; j < lend; ++j) {  if (c[i] == d[j]) {  counter++;  break;  }  }  }  if (counter == lenc) {return true;  }  }  return false;  }  int main() {  string a;  string b;  cout << "Enter a word : ";  cin >> a;  cout << "Enter another word : ";  cin >> b;  if(isanagram(a, b)){  cout<<"True"<<endl;  }  else{  cout<<"False"<<endl;  }  return 0;  } |
| int main(){  int a[10][10], b[10][10], mult[10][10], r1,  c1, r2, c2, i, j, k;  cout << "Enter row and col for 1stmatrix: ";  cin >> r1 >> c1;  cout << "Enter row and col for 2ndmatrix: ";  cin >> r2 >> c2;  while (c1!=r2){  cout << "Error!"<<endl;  cout <<"Enter row and col for 1stmatrix: ";  cin >> r1 >> c1;  cout <<"Enter row and col for 2ndmatrix: ";  cin >> r2 >> c2;  }  cout << endl << "Enter elements of matrix 1:" << endl;  for(i = 0; i < r1; ++i){  for(j = 0; j < c1; ++j){  cout << "Enter element a" << i + 1 << j + 1 << " : ";  cin >> a[i][j];  }  }  cout << endl << "Enter elements of matrix 2:" << endl;  for(i = 0; i < r2; ++i){  for(j = 0; j < c2; ++j){  cout << "Enter element b" << i + 1 << j + 1 << " : ";  cin >> b[i][j];  }  }  for(i = 0; i < r1; ++i){  for(j = 0; j < c2; ++j){  mult[i][j]=0;  }  }  for(i = 0; i < r1; ++i){  for(j = 0; j < c2; ++j){  for(k = 0; k < c1; ++k){  mult[i][j] += a[i][k] \* b[k][j];  }  }  }  cout << endl << "Output Matrix: " << endl;  for(i = 0; i < r1; ++i){  for(j = 0; j < c2; ++j){  cout << " " << mult[i][j];  if(j == c2-1){  cout << endl;  }  }  } return 0;  } | void dotProduct(int A[], int B[]) {  int sum =0;  cout << "Matrix A: ";  for (int i = 0; i <5; ++i) {  cout << A[i] << " ";  }  cout << endl;  cout << "Matrix B: ";  for (int i = 0; i < 5; ++i) {  cout << B[i] << " ";  }  cout << endl;  for (int i = 0; i < 5; ++i) {  sum += A[i] \* B[i];  }  cout << "Dot product is: " << sum << endl;  }  int main(){int A[5]={ 3, 4, 5, 6, 7 };  int B[5] ]={ 3, 4, 5, 6, 7 };  dotProduct(A,B);  return 0;  }  …………………………………………………………………………………………………...  void crossProduct(int a[], int b[],int arr[][5],int row,int col) {  for (int i = 0; i < row; i++) {  for (int j = 0; j < col; j++) {  arr[i][j] = a[i] \* b[j];  }  }  }  int main() {  const int row = 5;  const int col = 5;  int a[5] = { 3,4,5,6,7 };  int b[5] = { 3,4,5,6,7 };  int array[row][col];  crossProduct(a, b, array, row, col);  cout << "Cross product: " << endl;  for (int i = 0; i < row; i++) {  for (int j = 0; j < col; j++) {  cout << array[i][j] << " ";  }cout << endl;  }return 0;  } | //Checks whether one array is found in another  #include<cstdlib>  #include<ctime>  using namespace std;  bool contains(int smallArr[], int smallSize,  int bigArr[], int bigSize) {  for (int i = 0; i < bigSize; i++) {  int c = 0;  if (smallArr[0] == bigArr[i]) {  for (int j = 0; j < smallSize; j++) {  if (smallArr[j] == bigArr[i + j]){ c++;}  }  if (c == smallSize){return true;}  }  }return false;  }  int main() {  srand(time(0));  int smallArr[3], bigArr[11];  for (int i = 0; i < 3; i++){  smallArr[i] = rand() % 10;  }  for (int i = 0; i < 11; i++){  bigArr[i] = rand() % 10;  }  cout << "Small Array: ";  for (int i = 0; i < 3; i++){  cout << smallArr[i] << " ";  }  cout << endl << "Big Array: ";  for (int i = 0; i < 11; i++){  cout << bigArr[i] << " ";  }  cout << endl;  if (contains(smallArr, 3, bigArr, 11)) {  cout << "Yes" << endl;  }  else{  cout << "No" << endl;  } return 0;  }  ……………………………………………………………………………………  //frequency counter  int main() {  int arr[5] = { 1,4,4,6,7 };  for (int i = 0; i < 41; ++i){  int cnt = 0;  for(int j=0; j<5; j++){  if(arr[j]==i){ cnt++;  }  }  cout << i << " appears " << cnt <<" time."<< endl;  }return 0;  } |