REPORT #1

|  |
| --- |
| **제 출 일** : 2017년 09월 12일 |
| **과 목 명** : 컴퓨터프로그래밍및실습 |
| **담당교수 : 김정국 교수님** |
| **학 과** : 컴퓨터전자시스템 |
| **학 번** : 201703091 |
| **성 명 : 전기범** |

**HANKUK UNIVERSITY OF FOREIGN STUDIES**

****

**1 . 포인터 연산 예제**

* 1. **문제 정의**

**-> 자료형에 따른 포인터 값 증가의 차이를 알아보기 위한 문제**

**1.2. 문제 해결 방법**

**-> char, int, double형 변수를 선언하여 기존 값과 증가한 후의 값을 비교하였습니다.**

**1.3. 소스 코드**

|  |
| --- |
| #include <iostream>  using namespace std;  int main()  {  char \*pc;  int \*pi;  double \*pd;  pc = (char \*)10000;  pi = (int \*)10000;  pd = (double \*)10000;  cout << "증가 전 pc = " << (void \*)pc << " pi = " << pi << " pd = " << pd << endl;  pc++;  pi++;  pd++;  cout << "증가 후 pc = " << (void \*)pc << " pi = " << pi << " pd = " << pd << endl;  return 0;  } |

**1.4 결과 화면**

|  |
| --- |
| **C:\Users\dona9848\AppData\Local\Microsoft\Windows\INetCache\Content.Word\실행결과.png** |

**2 . 배열의 합과 평균 계산 프로그램**

**2.1 문제 정의**

**크기가 10인 1차원 배열의 값을 각각 입력받고, 배열의 모든 원소들의 합과 평균을 출력하도록 한다**

**2.2. 문제 해결 방법**

**10칸짜리 1차원 배열을 선언 후 내용을 입력받기 위해 cin사용, 합과 평균을 구하기 위해서 반복문을 사용하였으며, 값을 출력하기 위해서 cout을 사용하였습니다.**

**2.3. 소스 코드**

|  |
| --- |
| #include <iostream>  using namespace std;  int main()  {  int arr[10] = { 0 };  int sum = 0, avg = 0;  for (int loop = 0; loop < 10; loop++)  {  cin >> arr[loop];  if (loop == 9)  {  for (int num = 0; num < 10; num++)  {  sum += arr[num];  avg += arr[num];  }  avg /= 10;  }  }  cout << "배열의 합 = " << sum << endl;  cout << "배열의 평균 = " << avg << endl;  return 0;  } |

**2.4 결과 화면**

|  |
| --- |
| **C:\Users\dona9848\AppData\Local\Microsoft\Windows\INetCache\Content.Word\실행결과1.png**  **C:\Users\dona9848\AppData\Local\Microsoft\Windows\INetCache\Content.Word\실행결과3.png**  **C:\Users\dona9848\AppData\Local\Microsoft\Windows\INetCache\Content.Word\실행결과2.png** |

**3 . 행렬의 회전 프로그램**

**3.1 문제 정의**

[3][3], [5][5], [7][7]의 행렬의 값을 1씩 출력하되, 90도로 회전한 행렬을 출력한다.

**3.2. 문제 해결 방법**

3\*3, 5\*5, 7\*7 사이즈의 2차원 배열을 3개 생성 후 1로 초기화, for문을 이용하여 값을 채워 넣었습니다. 그 후 함수를 사용하여 배열의 값을 불러온 뒤, 배열의 위치를 변경하여 해당 문제를 해결하였습니다.

**3.3. 소스 코드**

|  |
| --- |
| #include <iostream>  using namespace std;  void mat9(int(\*arr)[3]);  void mat25(int(\*arr)[5]);  void mat49(int(\*arr)[7]);  int main(void)  {  int num = 1; // 배열의 값을 넣기 위해 사용  int mat3[3][3] = { 0 }, mat5[5][5] = { 0 }, mat7[7][7] = { 0 };  int col, row; // for문 제어 행,렬 변수  for (col = 0; col < 3; col++)  {  for (row = 0; row < 3; row++)  mat3[col][row] = num++;  }  num = 1; // 다음 출력을 위해 1로 초기화  for (col = 0; col < 5; col++)  {  for (row = 0; row < 5; row++)  mat5[col][row] = num++;  }  num = 1;  for (col = 0; col < 7; col++)  {  for (row = 0; row < 7; row++)  mat7[col][row] = num++;  }  mat9(mat3);  mat25(mat5);  mat49(mat7);  return 0;  }  void mat9(int(\*arr)[3]) // 배열 회전 및 출력 함수  {  int ary[3][3];  int col, row;  for (col = 0; col < 3; col++)  {  for (row = 0; row < 3; row++)  ary[col][row] = arr[col][row]; // 배열 복사  }  for (col = 0; col < 3; col++)  {  for (row = 0; row < 3; row++)  arr[col][row] = ary[row][2 - col]; // 회전  }  cout << endl;  for (col = 0; col < 3; col++)  {  for (row = 0; row < 3; row++)  cout << " " << arr[col][row] << " "; // 출력  cout << endl << endl;  }  }  void mat25(int(\*arr)[5])  {  int ary[5][5];  int col, row;  for (col = 0; col < 5; col++)  {  for (row = 0; row < 5; row++)  ary[col][row] = arr[col][row];  }  for (col = 0; col < 5; col++)  {  for (row = 0; row < 5; row++)  arr[col][row] = ary[row][4 - col];  }  cout << endl;  for (col = 0; col < 5; col++)  {  for (row = 0; row < 5; row++)  cout << " " << arr[col][row] << " ";  cout << endl << endl;  }  }  void mat49(int(\*arr)[7])  {  int ary[7][7];  int col, row;  for (col = 0; col < 7; col++)  {  for (row = 0; row < 7; row++)  ary[col][row] = arr[col][row];  }  for (col = 0; col < 7; col++)  {  for (row = 0; row < 7; row++)  arr[col][row] = ary[row][6 - col];  }  cout << endl;  for (col = 0; col < 7; col++)  {  for (row = 0; row < 7; row++)  cout << " " << arr[col][row] << " ";  cout << endl << endl;  }  } |

**3.4 결과 화면**

|  |
| --- |
|  |

**4 . 의 계산 결과 및 개수 출력 프로그램**

**4.1 문제 정의**

a와 b의 범위는 각각 2~5이다. 의 계산 결과 및 개수를 출력 하되, 중복된 수를 제외한 숫자를 나타내야 하며, 나타내는 순서 또한 오름차순으로 정렬해야한다.

**4.2. 문제 해결 방법**

**이중 for문을 이용하여** 의 계산을 표현하였으며, 계산을 하기 위해서 math.h해더파일 사용, pow함수를 이용하였습니다. 배열의 접근 용이성을 위해 전역변수로 설정하였으며, bubble sort를 사용하여 정렬하여 출력하였습니다.

**4.3. 소스 코드**

|  |
| --- |
| #include <iostream>  #include <math.h>  using namespace std;  int arr[16] = { 0 }; // 접근 용이성을 위해 전역변수 선언, 결과값 저장 배열  void print(); // 계산 출력  void sort(); // 정렬 후 출력  int main()  {  int a, b, ar[4] = { 2, 3, 4, 5 }, br[4] = { 2, 3, 4, 5 }, num = 0; // ar,br은 값 계산을 위한 배열  print();  for (a = 0; a < 4; a++) // 계산결과 넣기  for (b = 0; b < 4; b++)  {  arr[num] = pow(ar[a], br[b]);  num++;  }  sort();  cout << endl;  }  void print() // 출력 함수  {  int a, b;  for (a = 2; a < 6; a++)  {  for (b = 2; b < 6; b++)  {  cout << a << "^" << b << "=" << pow(a, b);  if (b == 5) // a^5 값 뒤에 , 출력 안하게 하기  {  continue;  cout << ", ";  }  else  cout << ", " << " ";  }  cout << endl; // a값 증가에 따른 줄 바꿈  }  cout << endl;  }  void sort() // 정렬 함수  {  int i, j, temp, count = 0;  for (i = 15; i >= 0; i--) // 정렬할 데이터의 범위  for (j = 0; j < i; j++) // 정렬할 데이터  if (arr[j] > arr[j + 1]) // 인접한 데이터 중 왼쪽이 더 크면  {  temp = arr[j]; // 인접한 데이터를 서로 바꾼다.  arr[j] = arr[j + 1];  arr[j + 1] = temp;  }  for (i = 0; i < 17; i++) // 카운팅  {  if (arr[i] == arr[i + 1])  continue;  else  count++;  }  cout << endl << "Count = " << count << endl;  for (i = 0; i < 17; i++) // 출력  {  if (arr[i] == arr[i + 1])  continue;  else  {  cout << arr[i];  if (arr[i] == 3125) // 마지막 값 뒤에 , 출력 안하게 하기  continue;  else  cout << ", ";  }    }  } |

**4.4 결과 화면**

|  |
| --- |
|  |