REPORT #4



|  |
| --- |
| **제 출 일** : 2017년 10월 10일 |
| **과 목 명** : 컴퓨터프로그래밍및실습 |
| **담당교수 : 김정국 교수님** |
| **학 과** : 컴퓨터전자시스템 |
| **학 번** : 201703091 |
| **성 명 : 전기범** |

**HANKUK UNIVERSITY OF FOREIGN STUDIES**



**1 . 문자 구분 프로그램**

* 1. **문제 정의**

사용자로부터 문자열을 입력 받은 후, 입력 받은 문자열에서 대문자, 소문자, 공백 문자, 그 외의 문자 수를 차례대로 출력한다.

**1.2. 문제 해결 방법**

Char형 배열을 선언, cin.getline을 사용하여 문자열을 입력받은 후, 반복문을 돌며 각 자리의 문자가 조건에 맞는 범위가 있는지 확인, 있다면 카운팅을 하여 마지막에 순서대로 출력했습니다.

**1.3. 소스 코드**

|  |
| --- |
| #include <iostream>  using namespace std;  int main()  {  char arr[10000]; // 문자열 입력 받을 배열  int Bcnt = 0, Lcnt = 0, Ncnt = 0, Scnt = 0; // 대문자, 소문자, 공백문자, 특수문자 카운트 변수  cin.getline(arr, 10000);  for (int i = 0; arr[i] != 0; i++) // NULL문자 까지 for문 반복  {  if ((arr[i] >= '!') && (arr[i] <= '@')) // 특수문자일경우  Scnt++;  else if ((arr[i] >= '[') && (arr[i] <= '`')) // 특수문자일경우  Scnt++;  else if ((arr[i] >= '{') && (arr[i] <= '~')) // 특수문자일경우  Scnt++;  else if ((arr[i] >= 'A') && (arr[i] <= 'Z')) // 대문자일경우  Bcnt++;  else if ((arr[i] >= 'a') && (arr[i] <= 'z')) // 소문자일경우  Lcnt++;  else if (arr[i] == ' ') // 공백일경우  Ncnt++;  }  cout << Bcnt << " " << Lcnt << " " << Ncnt << " " << Scnt << endl;  } |

**1.4 결과 화면**

|  |
| --- |
|  |

**2 . 마름모 출력 프로그램**

**2.1 문제 정의**

입력되는 크기 만큼의 마름모를 만드는데, 격 줄로 한 줄 씩 제거된 마름모를 만든다. 위에서 아래로, 왼쪽에서 오른쪽 순서로 대문자 알파벳을 이용하여 내용을 채운다.

**2.2. 문제 해결 방법**

숫자를 입력 받으면 /2+0.5를 진행하여 가운데 줄의 문자 출력 개수와 상단 마름모의 범위를 정했습니다. 또한 다시 계산한 숫자가 홀수일 경우와 짝수일 경우를 생각하여 상하단 출력을 위한 조건문을 생성하였고, 조건문안에서 공백 출력을 한 뒤 짝수 자리일 때 알파벳 출력을, 홀수 자리일 때는 공백을 출력하도록 하였습니다. 알파벳을 출력할 때 Z가 넘어가면 다시 A부터 출력하도록 조건문을 넣어주었습니다.

**2.3. 소스 코드**

|  |
| --- |
| #include <iostream>  using namespace std;  int main()  {  int num;  int a, b;  int i, cnt = 0;  char alp = 'A';  cin >> num;  num = (num / 2.0 + 0.5); // 상 하단 출력을 위한 수식  if (num % 2 == 0) // num이 짝수일때(7+4n)  {  for (i = 0; i < num + 1; i++) // 마름모 상단 출력  {  for (a = num - 1; a > cnt; a--) // 공백 출력  cout << " ";  for (b = 0; b <= cnt; b++)  {  if (b % 2 == 0) // 짝수 자리일때 출력  {  if (alp > 'Z')  alp = 'A';  cout << alp << " ";  alp++;  }  else // 홀수 자리일때 공백 출력  cout << " ";  }  cout << endl;  cnt++; // 제어 카운트 증가  if (cnt >= num) // 카운트가 숫자보다 커지면 반복문 종료  break;  }  }  else // num이 홀수일때 (5+4n)  {  for (i = 0; i < num + 1; i++) // 마름모 상단 출력  {  for (a = num - 1; a > cnt; a--) // 공백 출력  cout << " ";  for (b = 0; b <= cnt; b++)  {  if (b % 2 == 0) // 짝수 자리일때 출력  {  if (alp > 'Z')  alp = 'A';  cout << alp << " ";  alp++;  }  else // 홀수 자리일때 공백 출력  cout << " ";  }  cout << endl;  cnt++; // 제어 카운트 증가  if (cnt >= num) // 카운트가 숫자보다 커지면 반복문 종료  break;  }  }  int ncnt = num;  int mcnt = num - 1;  int dcnt = 0; // 하단부 출력을 위한 변수 선언  if (num % 2 == 0) // num이 짝수일때(7+4n)  {  for (i = 0; i < num - 2; i++) // 하단부 출력  {  int temp = i;  if (temp / 2 > 0 && temp % 2 == 0)  dcnt -= 4;  for (a = 3; a > dcnt; a--)  cout << " ";  int run = mcnt / 2;  while (run > 0)  {  if (alp > 'Z')  alp = 'A';  cout << alp;  alp++;  if (run != 1) // 그 줄의 마지막 문자 출력이 아닐경우  {  for (a = 0; a < 3; a++)  cout << " ";  }  else // 그 줄의 마지막 문자 출력일 경우  {  for (a = 0; a < 3; a++)  cout << " ";  }  run--;  }  cout << endl;  mcnt--;  dcnt++;  }  }  else // num이 홀수일때 (5+4n)  {  for (i = 0; i < num - 1; i++) // 하단부 출력  {  int temp = i;  if (temp / 2 > 0 && temp % 2 == 0)  dcnt -= 4;  for (a = 3; a > dcnt; a--)  cout << " ";  int run = ncnt / 2;  while (run > 0)  {  if (alp > 'Z')  alp = 'A';  cout << alp;  alp++;  if (run != 1) // 그 줄의 마지막 문자 출력이 아닐경우  {  for (a = 0; a < 3; a++)  cout << " ";  }  else // 그 줄의 마지막 문자 출력일 경우  {  if (num == 5) // 9 입력시 하단부 뒤에 공백 1개  for (a = 0; a < 1; a++)  cout << " ";  else // 다른수 일경우 공백 3개  for (a = 0; a < 3; a++)  cout << " ";  }  run--;  }  cout << endl;  ncnt--;  dcnt++;  }  }  } |

**2.4 결과 화면**

|  |
| --- |
|  |

**3 . 특정 범위에서의 오락가락 수 출력 프로그램**

**3.1 문제 정의**

**숫자를 왼쪽부터 오른쪽으로 읽어나갈 때, 오른쪽의 자리가 왼쪽의 자리 수 보다 작지 않다면 그 수를 증가수 라고 하며, 왼쪽 숫자보다 오른쪽 숫자가 크지 않으면 감소수 라고 한다. 증가수도 감소수도 아닌 나머지 수들을 오락가락 수 라고 하며 시작과 끝 범위를 입력, 그 범위 안에 있는 오락가락 수의 개수를 출력하는 프로그램이다.**

**3.2. 문제 해결 방법**

시작과 끝 범위를 입력, 반복문을 이용하여 배열을 초기화 시킨 후, 반복문을 통하여 해당 숫자의 오락가락수 여부를 판별하도록 하였습니다. 판별을 위하여 각 자리를 다른 배열에 저장, 왼쪽과 오른쪽 숫자의 증감, 동일여부를 확인, 일정 조건이 맞다면 카운트를 하였습니다.

1000~21780사이의 범위에서는 제외해야 하는 경우들이 있어 특정 조건이 맞다면 카운트가 줄어들도록 하여 오락가락수의 개수를 정확히 측정하도록 하였습니다.

**3.3. 소스 코드**

|  |
| --- |
| #include <iostream>  using namespace std;  void check();  int num[21781] = { 0 }; // 1 ~ 21780까지 숫자 넣어두는 배열  int arr[6]; // 입력받은 수를 한자리씩 비교하기 위한 배열  int main()  {  int start, end; // 시작 끝 값  cin >> start >> end;  for (int i = 0; i < end; i++) // 초기화  {  if (start > end) // 시작값이 끝 값보다 커지면 초기화 끝냄  break;  num[i] = start++; // 시작값 ~ 끝값 초기화  }    check();  /\*for (int i = 0; i < end; i++)  cout << num[i] << " ";\*/  /\*for (int i = 0; i < 6; i++)  cout << arr[i] << " ";\*/  }  void check() // 오락가락수 체크 및 출력 함수  {  int cnt1 = 0, cnt2 = 0, cnt3 = 0;// cnt1 100 ~ 999사이 오락가락,cnt2 1000 ~ 9999사이 오락가락, cnt3 10000 ~ 21780사이 오락가락  for (int i = 0; num[i] != 0; i++)  {  int bcnt = 0, lcnt = 0, scnt = 0; // bcnt 증가수 lcnt 감소수 scnt 동일숫자 판단 카운트  if (num[i] > 99 && num[i] < 1000) // 100 ~ 999  {  arr[0] = num[i] / 100 % 100;  arr[1] = num[i] / 10 % 10;  arr[2] = num[i] % 10;  for (int j = 0; j < 2; j++) // 오락가락수 확인  {  if (arr[j] > arr[j + 1])  bcnt++;  else if (arr[j] == arr[j + 1])  scnt++;  else if (arr[j] < arr[j + 1])  lcnt++;  }  if (bcnt < 2 && lcnt < 2 && scnt < 1)  {  cnt1++;  //cout << num[i] << " ";  }  }  else if (num[i] > 999 && num[i] < 10000) // 1000 ~ 9999  {  arr[0] = num[i] / 1000 % 10;  arr[1] = num[i] / 100 % 10;  arr[2] = num[i] / 10 % 10;  arr[3] = num[i] % 10;  for (int j = 0; j < 3; j++) // 오락가락수 확인  {  if (arr[j] > arr[j + 1])  bcnt++;  else if (arr[j] == arr[j + 1])  scnt++;  else if (arr[j] < arr[j + 1])  lcnt++;  }  if (bcnt <3 && lcnt <3 && scnt <2)  {  cnt2++;  //cout << num[i] << " ";  }  if (bcnt == 0 && lcnt == 2 && scnt == 1) // 잘못 카운트 된 숫자들 제거  cnt2--;  else if (bcnt == 2 && lcnt == 0 && scnt == 1)  cnt2--;  }  else if (num[i] > 9999 && num[i] <= 21780) // 10000 ~ 21780  {  arr[0] = num[i] / 10000 % 10;  arr[1] = num[i] / 1000 % 10;  arr[2] = num[i] / 100 % 10;  arr[3] = num[i] / 10 % 10;  arr[4] = num[i] % 10;  for (int j = 0; j < 4; j++) // 오락가락수 확인  {  if (arr[j] > arr[j + 1])  bcnt++;  else if (arr[j] == arr[j + 1])  scnt++;  else if (arr[j] < arr[j + 1])  lcnt++;  }  if (bcnt < 4 && lcnt < 4 && scnt < 3)  {  cnt3++;  //cout << num[i] << " ";  }  if (bcnt == 0 && lcnt == 3 && scnt == 1)  cnt3--;  else if (bcnt == 3 && lcnt == 0 && scnt == 1)  cnt3--;  else if (bcnt == 0 && lcnt == 2 && scnt == 2)  cnt3--;  else if (bcnt == 2 && lcnt == 0 && scnt == 2)  cnt3--;  }  }    cout << cnt1 + cnt2 + cnt3 << endl;  } |

**3.4 결과 화면**

|  |
| --- |
|  |

**4 . 특정 범위에서의 a^b의 갯수 출력 프로그램**

**4.1 문제 정의**

2<=a<=100, 2<=b<=100의 범위 내에서 입력을 받아 a^b의 갯수를 출력하는 프로그램

**4.2. 문제 해결 방법**

이전 과제의 소스코드를 참조하였으며, a와b의 값을 입력, 이중for문을 도는데 횟수 제어는 입력 받은 수의-1 미만까지 돌도록 하였습니다. 그 이유로는 시작 값이 2이기 때문에 2+i를 진행하여 입력 받은 수 만큼까지 거듭제곱을 진행하기 위함입니다.

Pow를 사용하여 제곱을 계산하고 arr이라는 전역변수에 넣어 두었습니다. 이때 arr배열은 int형이 아닌 double형으로 했는데요, a와b의 최대값이 100이라 100^100=1.e+200 이라는 결과가 나와 double형이 10^-308~10^308(1.e+308)으로 가장 큰 표현이 가능하여 double형을 사용하였습니다. 그 후 버블 정렬을 사용하여 정렬, 옆 숫자와 비교하여 다르다면 카운트를 올리고 같다면 넘어가도록 하여 갯수를 출력하도록 하였습니다. 이때 버블정렬에서 사용한 temp변수 또한 double형을 사용해 앞의 arr배열의 값 손실 및 범위를 만족시켰습니다.

**4.3. 소스 코드**

|  |
| --- |
| #include <iostream>  #include <cmath>  using namespace std;  double arr[10000] = { 0, }; // 접근 용이성을 위해 전역변수 선언, 결과값 저장 배열  void sort(int num); // 정렬 후 출력  int main()  {  int a, b;  int num = 0;  cin >> a >> b; // 끝 범위 입력 받기  for (int i = 2; i <= a; i++) // 계산결과 넣기  for (int j = 2; j <= b; j++)  {  arr[num] = pow(i, j);  num++;  }  sort(num);  cout << endl;  /\*for (int i = 0; i < num; i++)  cout << arr[i] << " ";\*/  }  void sort(int num) // 정렬 함수  {  int i, j, count = 0;  double temp;  for (i = num - 1; 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 < num; i++) // 카운팅  {  if (arr[i] == arr[i + 1])  continue;  else  count++;  }  cout << count << endl;  } |

**4.4 결과 화면**

|  |
| --- |
|  |