REPORT #4

|  |
| --- |
| **제 출 일** : 2017년 10월 10일 |
| **과 목 명** : 컴퓨터 프로그래밍 및 실습 |
| **담당교수 : 김정국 교수님** |
| **학 과** : 컴퓨터 전자 시스템 공학부 |
| **학 번** : 201702234 |
| **성 명 : 유동혁** |

**HANKUK UNIVERSITY OF FOREIGN STUDIES**

****

**1. 문자 구분 프로그램**

* 1. **문제 정의**

-> 문장을 구성하는 문자들을 분류함

**1.2 문제 해결 방법**

-> 아스키 코드를 이용해 각 개수를 배열에 저장함

**1.3 소스 코드**

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {    char munjang[10000] = {0};  int count[4] = {0};  int x;    cin.getline(munjang, 10000);    //아스키 코드 표 공백: 32; 대문자: 65~90; 소문자: 97~122;  for(x=0; munjang[x]!=0; x++)  {  if(munjang[x] >= 65 && munjang[x] <= 90)  count[0]++;  else if(munjang[x] >= 97 && munjang[x] <= 122)  count[1]++;  else if(munjang[x] == 32)  count[2]++;  else  count[3]++;  }    for(x=0; x<4; x++)  cout << count[x] << " ";    return 0;  } |

**1.4 결과 화면**

|  |
| --- |
|  |

**2. 격자 마름모 출력 프로그램**

**2.1 문제 정의**

-> 격자로 마름모를 출력하는 프로그램

**2.2 문제 해결 방법**

-> 각 층에 출력해야 하는 글자 수를 계산해(= 2 \* (층)) 알파벳과 공백을 출력한다. 만약 알파벳의 범위를 넘어가면 다시 A로 돌아가게 한다.

\*\*\*25를 넣었을 때까지는 정상출력 되지만, 27이상의 숫자를 넣으면 멈추지않고 무한히 출력한다. 범위를 설정하지 않았는데도 이런 문제가 생기는 이유는 무엇일까?

**2.3 소스 코드**

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  char alpha = 65;  int x, y, z, input;  int count;  cin >> input;  //마름모의 상단  for(x=1; x<=input/2; x++)  {  count = 0;  //마름모를 구성 하기 전 공백 출력  for(y=1; y<=input/2-x+1; y++)  cout << " ";  //마름모 구성  while(1)  {  cout << alpha;  alpha++;  //알파벳을 출력하고나서 다음 알파벳을 준비 시킨다  count++;  if(alpha == 91)  alpha = 65;  //만약 알파벳 범위를 넘어가면 다시 A로 돌아감  if(count == 2\*x-1)  goto next1;  //goto문: 각 층에 들어가야 하는 글자 수를 충족하면 그 층의 출력을 종료하고 다음 층으로 넘어감  for(z=1; z<=3; z++)  {  cout << " ";  //한번에 출력되는 공백의 갯수는 최대 3개  count++;  if(count == 2\*x-1)  goto next1;  }  }  next1:  cout << " " << endl;  }  //마름모의 가운데  count = 0;  while(1)  {  cout << alpha;  alpha++;  count++;  if(alpha == 91)  alpha = 65;  if(count == input)  goto next3;  for(z=1; z<=3; z++)  {  cout << " ";  count++;  if(count == input)  goto next3;  }  }  next3:  cout << " " << endl;  //마름모의 하단  for(x=1; x<=(input-1)/2; x++)  {  if(input%4 == 3)  {  if(x == input/2)  break;  }  //층 수를 4로 나눈 나머지가 3일때는 마지막 층에 공백만 출력되기 때문에 그 줄을 출력 안되게 함  count = 0;  for(y=1; y<=x; y++)  cout << " ";  if(x%2 == 1)  {  cout << " ";  count += 2;  }  while(1)  {  cout << alpha;  alpha++;  count++;  if(alpha == 91)  alpha = 65;  if(count >= input - 2\*x)  goto next2;  for(z=1; z<=3; z++)  {  cout << " ";  count++;  if(count >= input - 2\*x)  goto next2;  }  }  next2:  cout << " " << endl;  }  return 0;  } |

**2.4 결과 화면**

|  |
| --- |
|  |

**3. 오락가락수 개수 출력 프로그램**

**3.1 문제 정의**

-> 증가 수도, 감소 수도 아닌 오락가락수의 개수를 세서 출력해주는 프로그램이다.

**3.2 문제 해결 방법**

-> 증가 수와 감소 수의 개수를 모두 세서 범위 내 전체 수의 개수에서 뺐다.

**3.3 소스 코드**

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  int su[5] = {0};  int x, y;  int i, k;  int count = 0;    cin >> x >> y;  for(i=x; i<=y; i++)//x부터 y사이에서 변함  {  int n = i;  for(k=4; k>=0; k--)  {  su[k] = n%10;  n /= 10;  }  //각 자리수를 배열에 저장  if(su[0] == 0)  {  if(su[1] == 0)  {  if(su[2] == 0)//2자리 수면  {  if(su[3] >= su[4])  count++;  else if(su[3] <= su[4])  count++;  }  else//3자리 수면  {  if(su[2] >= su[3] && su[3] >= su[4])  count++;  else if(su[2] <= su[3] && su[3] <= su[4])  count++;  }  }  else//4자리 수면  {  if(su[1] >= su[2] && su[2] >= su[3] && su[3] >= su[4])  count++;  else if(su[1] <= su[2] && su[2] <= su[3] && su[3] <= su[4])  count++;  }    }  else//5자리 수면  {  if(su[0] >= su[1] && su[1] >= su[2] && su[2] >= su[3] && su[3] >= su[4])  count++;  else if(su[0] <= su[1] && su[1] <= su[2] && su[2] <= su[3] && su[3] <= su[4])  count++;  }  }  cout << (y-x+1) - count << endl;//숫자의 총 갯수에서 증가수, 감소수를 뺀 값 = 오락가락수 갯수 출력  return 0;  } |

**3.4 결과 화면**

|  |
| --- |
|  |

**4. 특정 범위에서 a^b의 개수 출력 프로그램**

* 1. **문제 정의**

-> 2^2, 2^3, …, 2^b, 3^2, …, 3^b, …, a^b의 결과의 가짓수를 모두 구한다

**4.2 문제 해결 방법**

-> 가능한 모든 경우의 수의 결과값을 배열에 저장하고 그 배열에 같은 값이 존재한다면 방금 저장한 값을 0으로 바꾼다. 그리고, 배열을 처음부터 끝까지 검토해 0이 아닌 값의 개수를 센다(a^b의 형태에서 0이라는 결과값이 나올 수 없기 때문이다.).

**4.3 소스 코드**

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {    int a, b;  int x, y, z;  int tmp = 0, i;  int count=0;  double arr[100000] = {0};    cin >> a >> b;  if(a>=2 && a<=100 && b>=2 && b<=100)  {  for(x=2; x<=a; x++)//x는 2부터 a까지 변한다  {  for(y=2; y<=b; y++)//y는 2부터 b까지 변한다  {  arr[tmp] = 1;//현재 배열을 1로 초기화  for(z=1; z<=y; z++)  arr[tmp] \*= x;  //x^y을 현재 배열에 저장  for(i=0; i<tmp; i++)  {  if(arr[tmp] == arr[i])  arr[tmp] = 0;  //arr[0]부터 arr[tmp-1]까지 arr[tmp]와 같은게 있다면 arr[tmp]를 0으로 바꿈  }  tmp++;  }  }  }    for(i=0; i<100000; i++)  {  if(arr[i] != 0)  count++;  }  //a^b의 결과 중에 0이 나올수 없으므로 0이 아닌 결과의 수를 센다  cout << count << endl;  return 0;  } |

**4.4 결과 화면**

|  |
| --- |
|  |