REPORT #2

|  |
| --- |
| **제 출 일** : 2017년 09월 19일 |
| **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. 문자열 길이 출력 프로그램**

* 1. **문제 정의**

-> 사용자로부터 문자열을 입력 받아 포인터를 이용하여 문자열의 길이를 출력한다.

**1.2 문제 해결 방법**

-> 배열의 모든 원소를 0으로 초기화 한 후, 첫번째 원소(=geul[0])의 포인터를 설정하고 반복문을 이용해 문자열의 끝(=0인 원소의 직전 원소)을 찾아 2번째 포인터를 설정한다.

그리고 (문자열의 마지막 글자의 주소) – (문자열의 첫번째 글자의 주소) + 1을 하면 글자 수가 나온다.

🡺 이 방법을 이용해 제출하면 100점이라고 뜨지않는다. 어느 부분이 문제일까?

**1.3 소스 코드**

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {    int x;  char geul[100] = {0};//배열 선언 & 초기화  char \*p1 = &geul[0];//0번째 배열의 포인터 설정    cin >> geul;    for(x=0; x<100; x++){  if(geul[x] == 0){//만약 배열의 원소가 0이면  char \*p2 = &geul[x-1];//0인 원소 직전의 원소(=문자열의 마지막글자)의 포인터 설정  cout << p2 - p1 +1;//문자열의 마지막 글자의 주소 - 문자열의 첫번째 글자 주소 + 1 = 글자수  break;//반복문 탈출  }  }  return 0;  } |

**1.4 결과 화면**

|  |
| --- |
|  |

**2. 문자의 빈도 출력 프로그램**

**2.1 문제 정의**

-> 사용자로부터 입력 받은 문자열의 각 알파벳의 빈도 수를 출력하는 프로그램

**2.2 문제 해결 방법**

-> 문자열의 공백도 입력 받을 수 있도록 cin.getline함수를 이용한다. 그리고 아스키코드에서 a~z는 97~122이므로 문자 빈도를 저장하는 배열은 크기를 123으로 하고, 각 알파벳에 대해 문자열을 처음부터 끝까지 비교하고 문자열에 해당 알파벳이 존재하면 그 문자의 빈도를 +1한다.

**2.3 소스 코드**

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  char geul[100] = {0};//문자열을 저장하는 배열  int alpha[123] = {0};//알파벳의 빈도를 저장하는 배열    char \*p;//문자열을 지정하는 포인터  int x, y, n=0;  char k;//아스키 코드를 이용하기 위한 변수    cin.getline(geul, 100);//공백을 포함하여 문자열을 입력 받을 수 있는 함수  p = geul;//포인터 초기화    for(x=0; x<100; x++)  {  if(p[x] == 0)  break;  else  n++;  }  //문자열의 글자 수를 계산한다  for(k=97; k<=122; k++)//아스키 코드에서 97~122 = a~z  {  for(y=0; y<n; y++)  {  if(p[y] == k)  alpha[k]++;  }  }  //각 알파벳에 대해 문자열을 검사한 후 빈도에 저장한다  for(k=97; k<=122; k++)  cout << k << " : " << alpha[k] << endl;  //모든 알파벳의 빈도를 출력  return 0;  } |

**2.4 결과 화면**

|  |
| --- |
|  |

**3. 문자열 공백 구분 출력 프로그램**

**3.1 문제 정의**

-> 문자열의 공백을 ‘줄 바꿈’으로 바꿔 출력하는 프로그램

**3.2 문제 해결 방법**

-> 문자열을 입력 받을 때 공백도 저장할 수 있는 cin.getline함수를 이용해 배열을 입력 받고, 별도로 반복문을 이용하여 특정 문자를 만나면 줄 바꿈을 하는 함수를 만들어 이용한다.

**3.3 소스 코드**

|  |
| --- |
| #include <iostream>  using namespace std;  void splitString(char\* str, char delim);  int main() {    char str[100] = {0};//배열 str 선언 & 배열의 모든 원소 0으로 초기화    cin.getline(str, 100);//공백을 포함한 문자열을 받을 수 있는 함수이다.    splitString(str, ' ');//문자열과 공백을 대입함  return 0;  }  void splitString(char\* str, char delim){    int x;    for(x=0; x<100; x++){  if(str[x] == 0)//배열의 원소가 0이면  break;//더이상 출력하지 않고 종료    if(str[x] == delim)//만약 배열의 원소가 delim일 때(여기서는 공백)  cout << endl;//줄바꿈  else//delim이 아니면  cout << str[x];//정상적으로 출력  }  } |

**3.4 결과 화면**

|  |
| --- |
|  |

**4. Palindrome 판단 프로그램**

* 1. **문제 정의**

-> 사용자로부터 입력 받은 문자열이 Palindrome(앞으로 읽어도 뒤로 읽어도 똑같은 문자열)인지 아닌지 판단해주는 프로그램

**4.2 문제 해결 방법**

-> 문자열의 글자 수를 계산하고 글자 수가 n일 때 반복문에서 x번째 원소와 n – x – 1번째 원소가 같은지를 비교하고,(ex. 글자수가 6일 때: 0번째 원소와 5번째원소, 1번째 원소와 4번째 원소, …) 만약 비교 하는 도중 규칙에 맞지 않음을 발견했을 때 ‘거짓’을 출력하고 함수를 종료한다. 만약 비교가 정상적으로 모두 이루어졌을 때(=규칙에 모두 맞을 때), ‘참’을 출력하고 함수를 종료한다.

**4.3 소스 코드**

|  |
| --- |
| #include <iostream>  using namespace std;  int isPalindrome(char\* str);  int main() {  char str[100] = {0};//문자열 선언 & 초기화  cin >> str;//사용자로부터 문자열을 입력 받음  isPalindrome(str);//함수에 문자열 대입  return 0;  }  int isPalindrome(char\* str){    int x, n=0;    for(x=0; x<100; x++){  if(str[x] == 0)  break;  else  n++;//배열의 원소 수 (==문자열의 글자수)를 계산  }  for(x=0; x<=n/2; x++){  if(str[x] != str[n-x-1]){//문자열을 반으로 나눴을 때 왼쪽 글자와 오른쪽 글자가 일치하지않으면  cout << "not palindrome";//"거짓"을 출력하고  return 0;//함수 종료  }    cout << "palindrome";//for문이 정상적으로 끝나면(중간에 함수가 종료되지 않았으면)"참"을 출력  return 0;  }  } |

**4.4 결과 화면**

