REPORT #2



|  |
| --- |
| **제 출 일** : 2017년 09월 19일 |
| **과 목 명** : 컴퓨터프로그래밍및실습 |
| **담당교수 : 김정국 교수님** |
| **학 과** : 컴퓨터전자시스템 |
| **학 번** : 201703091 |
| **성 명 : 전기범** |

**HANKUK UNIVERSITY OF FOREIGN STUDIES**



**1 . 문자열 길이 출력 프로그램**

* 1. **문제 정의**

사용자로부터 문자열을 입력 받고, 포인터를 사용하여 문자열의 길이를 계산, 그 문자열의 길이를 출력하는 프로그램

**1.2. 문제 해결 방법**

256칸의 char형 배열을 생성, cin.getline을 사용하여 해당 배열에 문자열을 입력받은 후 str이라는 함수에서 포인터로 문자열을 받은 후, if문을 사용하여 null문자 도달 시 종료, 그렇지 않으면 num이란 변수의 값을 1씩 증가시켜 문자열의 길이를 계산하게 하였습니다.

**1.3. 소스 코드**

|  |
| --- |
| #include <iostream>  using namespace std;  int num = 0; // 문자열 길이 계산 변수  void str(char \*ary);  int main()  {  char arr[256]; // 입력 받을 문자열  cin.getline(arr, 256);  str(arr);  cout << num << endl;  }  void str(char \*ary) // 포인터로 문자열 받고 길이 계산  {  int i;  for (i = 0; i < 256; i++)  {  if (ary[i] == '\0') // \0인 null문자 도달 시 중지  break;  else  num++;  }  } |

**1.4 결과 화면**

|  |
| --- |
|  |

**2 . 문자의 빈도 출력 프로그램**

**2.1 문제 정의**

사용자로부터 문자열을 입력 받은 후, 포인터를 이용하여 각 문자의 빈도를 계산하여 출력한다**.**

**2.2. 문제 해결 방법**

check이라는 함수 안에 alpha라는 26칸의 배열을 생성, 그 안에 알파벳 a~z까지 넣어둔 후, num이라는 26칸의 배열을 생성, 0으로 초기화 합니다. 포인터로 넘겨받은 str배열을 처음부터 끝까지 확인을 하며 같은 문자가 있을경우 num의 값을 1씩 증가 시켜 문자의 빈도를 계산하도록 만들었습니다**.**

**2.3. 소스 코드**

|  |
| --- |
| #include <iostream>  using namespace std;  void check(char \*ary);  int main()  {  char str[256];  cin.getline(str, 256);  check(str);  }  void check(char \*ary) // 문자 갯수 확인 함수  {  int i;  char alpha[26];  int num[26];  for (i = 0; i < 26; i++)  {  alpha[i] = 'a' + i;  num[i] = 0;  }  for (i = 0; ary[i] != 0; i++) // 문자열 끝까지 각각 확인  {  if ((ary[i] >= 'a') && (ary[i] <= 'z'))  num[ary[i] - 'a']++;  }  for (i = 0; i < 26; i++)  cout << alpha[i] << " : " << num[i]<<endl;  } |

**2.4 결과 화면**

|  |
| --- |
|  |

**3 . 문자열 공백 구분 출력 프로그램**

**3.1 문제 정의**

사용자로부터 입력 받은 문자열의 공백을 구분하여 공백일 경우 다음 줄에 표시하게 하는 프로그램

**3.2. 문제 해결 방법**

splitString이라는 함수에서 if문을 사용하여 null문자 도달 시 중지 하도록 만들었으며, delim이라는 조건과 같은 경우 다음 줄로 줄 바꿈 하여 출력하도록 만들었습니다. (여기서의 조건은 공백 문자일 경우)

**3.3. 소스 코드**

|  |
| --- |
| #include <iostream>  using namespace std;  void splitString(char \*str, char delim);  int main()  {  char str[256]; // 입력받을 문자열  cin.getline(str, 256);  splitString(str, ' ');  cout << endl;  }  void splitString(char \*ary, char delim) // 포인터로 문자열 받고 공백시 다음 줄 출력  {  int i;  for (i = 0; i < 256; i++)  {  if (ary[i] == '\0') // \0인 null문자 도달시 중지  break;  else if (ary[i] == delim) // 조건과 같을 경우 다음줄로 줄바꿈  cout << endl;  else  cout << ary[i];  }  } |

**3.4 결과 화면**

|  |
| --- |
|  |

**4 . Palindrome 판단 프로그램**

**4.1 문제 정의**

입력 받은 문자열이 앞으로 읽었을 때와 뒤로 읽었을 때 동일한 숫자 또는 문자열인 Palindrome인지 아닌지 판단하는 프로그램

**4.2. 문제 해결 방법**

포인터 2개를 사용하여 입력 받은 문자열의 처음의 주소와 마지막 주소를 받아 둘의 차는 문자열의 길이를 측정, 그 문자열의 절반만 for문을 돌게 하였으며 현재 돌고 있는 for문의 문자열 속 문자와 (문자열의 길이 - i – 1)번째의 문자와 비교, 같으면 카운트를 1씩 증가 시키고 같지 않다면 유지를 하여 해당 카운트와 문자열의 길이/2 가 같다면 회문으로 판단, 회문 여부를 출력하게 하였습니다

**4.3. 소스 코드**

|  |
| --- |
| #include<iostream>  using namespace std;  int len = 0; // len = arr 문자열의 길이 접근을 위하여 전역변수로 선언  int isPalindrome(char\* arr);  int main()  {  char str[256];  char \*ps = 0, \*pe =0; // \*ps는 문자열의 시작 주소, \*pe는 문자열의 끝 주소  cin >> str;  ps = &str[0];  for (int i=0;i<256;i++)  if (str[i]=='\0')  {  pe = &str[i];  break;  }  len = pe - ps;  isPalindrome(str);  }  int isPalindrome(char\* arr) // 회문 판단 프로그램  {  int i, count = 0; // count는 회문 여부 판단 변수  for (i = 0; i < len / 2; i++) // len의 절반만 for문 돎  {  if (arr[i] != arr[len - i - 1]) // 문자열의 길이-i-1과 현재 판단중인 문자열의 i번째가 같지 않다면 카운트 안올라감  count = count;  else if (arr[i] == arr[len - i - 1]) // 같다면 카운트 1증가  count++;  }  if (count != len / 2) // count가 len/2 와 같지 않다면 회문 아님  cout << "not palindrome" << endl;  else if (count == len / 2) // count와 len/2가 같다면 회문  cout << "palindrome" << endl;  return 0;  } |

**4.4 결과 화면**

|  |
| --- |
|  |