**자료구조 과제 3**

(Pattern Matching 실험)

logo.gif

|  |  |
| --- | --- |
| 과목명 | 자료구조 |
| 담당교수 | 김계영 |
| 학과 | 소프트웨어학부 |
| 학년 | 3 |
| 학번 | 20152994 |
| 성명 | 이진영 |
| 제출일 | 2017.5.31 |



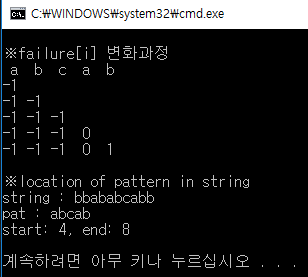
**과제3 - Pattern Matching 실험**

20152994 이진영

**1. 원시 코드**

|  |
| --- |
| #include <stdio.h>  #include <string.h>  #define max\_string\_size 100  #define max\_pattern\_size 100  int pmatch(char\* string, char\* pat);  void fail(char\* pat);  int failure[max\_pattern\_size];  char string[max\_string\_size] = "bbababcabb";  char pat[max\_pattern\_size] = "abcab";  void main() {  fail(pat);  printf("\n※location of pattern in string\n");  printf("string : %s\n", "bbababcabb");  printf("pat : %s\n", "abcab");  printf("start: %d, end: %d\n\n", pmatch(string, pat), pmatch(string, pat) + strlen(pat) - 1);  }  int pmatch(char\* string, char\* pat)  {  int i = 0, j = 0;  int lens = strlen(string);  int lenp = strlen(pat);  while (i<lens && j<lenp) {  if (string[i] == pat[j]) {  i++; j++;  }  else if (j == 0) i++;  else j = failure[j - 1] + 1;  }  return ((j == lenp) ? (i - lenp) : -1);  }  void fail(char\* pat) {  int i = 0, j = 0;  int n = strlen(pat);  failure[0] = -1;  printf("\n※failure[i] 변화과정\n");  printf(" a b c a b\n");  printf("%2d\n", failure[0]);  for (j = 1; j < n; j++) {  i = failure[j - 1];  while ((pat[j] != pat[i + 1]) && (i >= 0))  i = failure[i];  if (pat[j] == pat[i + 1])  failure[j] = i + 1;  else failure[j] = -1;  for (int i = 0; i <= j; i++) {  printf("%2d ", failure[i]);  }  printf("\n");  }  } |

**2. 실행 결과**

****