|  |  |  |  |
| --- | --- | --- | --- |
| 2022/2 『자료구조』과제 보고서 | | | |
| 제목 | 4장 실습( ) 과제( O ) | 제출일자 | 2022.  10 .    13 . |
| 학번 | 201911608 | 이름 | 김지환 |

|  |  |
| --- | --- |
| 연습문제 4장  02. 문자 A, B, C, D, E를 스택에 넣었다가 다시 꺼내어 출력하면 어떻게 되는가?  (1) A, B, C, D, E (2) E, D, C, B, A (3) A, B, C, E, D (4) B, A, C, D, E  --> stack={A,B,C,D,E}에서 LIFO 이므로 E, D, C, B, A 순서가 된다.  04. 배열로 구현된 스택에서 top이 3이면 현재 스택에 저장된 요소들의 개수는?  (1) 1 (2) 2 (3) 3 (4) 4  -> top의 위치는 요소의 개수 -1이다. 즉, top이 3이면 0, 1, 2, 3으로 요소의 개수 +1개로 총 4개의 요소가 있다.  06. 스택에 항목들을 삽입하고 삭제하는 연산은 시간 복잡도가 어떻게 되는가?  삽입 연산은 s->data[++(s->top)] = item; 으로 ++ 가산 1회, 전치연산 1회로 총 2회이다. 시간 복잡도는 상한 값이므로 O(1)의 시간 복잡도를 가진다.  삭제 연산은 s->data[(s->top)--]; 으로 -- 감산 1회로 시간 복잡도는 O(1)을 가진다.  08. 크기가 5인, 배열로 구현된 스택 A에 다음과 같이 삽입과 삭제가 되풀이되었을 경우에 각 단계에서 스택의 내용(1차원 배열의 내용, top의 값)을 나타내시오.  push(A, 1); // A = {1}, top = 0  push(A, 2); // A = {1, 2}, top = 1  push(A, 3); // A = {1, 2, 3}, top = 2  pop(A); // A = {1, 2}, top = 1  push(A, 4); // A = {1, 2, 4}, top = 2  push(A, 5); // A = {1, 2, 4, 5}, top = 3  pop(A); // A = {1, 2, 4}, top = 2  10. 배열에 들어 있는 정수의 순서를 거꾸로 하는 프로그램을 작성해보자. 스택을 사용한다.  실행결과   |  | | --- | | 정수 배열의 크기 : 6  정수를 입력하시오 : 1 2 3 4 5 6  반전된 정수 배열 : 6 5 4 3 2 1 |   #define \_CRT\_SECURE\_NO\_WARNINGS  #include <stdio.h>  #include <malloc.h>  int size;  typedef struct {  int\* data;  int top;  }Stack;  void push(Stack\* s, int item) {  if (size <= s->top) {  printf("FULL\n");  return;  }  s->data[++(s->top)] = item;  }  int pop(Stack\* s) {  if (0 > s->top) {  printf("EMPTY\n");  return -1;  }  return s->data[(s->top)--];  }  int main() {  int n;  printf("정수 배열의 크기 : ");  scanf("%d", &size);  Stack s = { malloc(sizeof(int) \* size), -1 };  printf("정수를 입력하시오 : ");  for (int i = 0;i < size;i++) {  scanf("%d", &n);  push(&s, n);  }  printf("반전 된 정수 배열 : ");  while (s.top > -1) {  int p = pop(&s);  if (p == -1) break;  printf("%d ", p);  }  }  실행결과 |
| 12. 다음과 같이 문자열을 압축하는 프로그램을 작성하라. “4a3b”는 ‘a’가 4개, ‘b’가 3개 있다는 의미이다. 이러한 압축 방법을 런길이(run length) 압축이라고 한다. 소문자와 대문자는 구별하지 않는다. 압축된 문자열에서는 소문자로 출력한다. 스택의 peek() 연산을 고려해보자.  실행결과   |  | | --- | | 문자열을 입력하시오 : aaaAbBb  압축된 문자열: 4a3b |   #define \_CRT\_SECURE\_NO\_WARNINGS  #include <stdio.h>  #include <malloc.h>  int len;  typedef struct {  int\* data;  int top;  }Stack;  void push(Stack\* s, int item) {  if (len <= s->top) {  printf("FULL\n");  return;  }  s->data[++(s->top)] = item;  }  int pop(Stack\* s) {  if (0 > s->top) {  printf("EMPTY\n");  return -1;  }  return s->data[(s->top)--];  }  int main() {  int ascii[92]={0,};  char str[101];  printf("문자열을 입력하시오 : "); gets(str);  len = strlen(str);  Stack s = { malloc(sizeof(int) \* len), -1 };    for (int i = 0; i < len; i++) push(&s, (int)str[i]);  int asc;  while (s.top != -1) {  asc = pop(&s);  if (asc >= 91) asc -= 32;  ascii[asc]++;  }  for (int i = 65; i < 91; i++)  if (ascii[i]) printf("%d%c", ascii[i], (char)(i+32));  }  실행결과    -> push와 pop을 이용 |
| 14. 배열로 구현된 스택에 저장된 요소의 수를 반환하는 size 연산을 구현하여 보라.  #include <stdio.h>  typedef struct {  int data[10];  int top;  }Stack;  void push(Stack\* s, int item) {  if (10 <= s->top) {  printf("FULL\n");  return;  }  s->data[++(s->top)] = item;  }  int size(Stack\* s) {  return (s->top)+1;  }  int main() {  Stack s = {{0},-1};  for (int i = 0;i < 5;i++) push(&s, i + 1);  printf("size - %d\n", size(&s));  }  실행결과    -> for문을 5번 돌렸으므로 size는 5가 맞다. |
| 16. 회문(palindrome)이란 앞뒤 어느 쪽에서 읽어도 같은 단어를 의미한다. 예를 들면 “eye”, “madam, I’m adam”, “race car”등이다. 여기서 물론 구두점이나 스페이스, 대소문자 등은 무시하여야 한다. 스택을 이용하여 주어진 문자열이 회문인지 아닌지를 결정하는 프로그램을 작성하라.  실행결과   |  | | --- | | 문자열을 입력하시오: madam  회문입니다. |   #include <stdio.h>  int len=0;  typedef struct {  char data[101];  int top;  }Stack;  void push(Stack\* s, int item) {  if (len <= s->top) {  printf("FULL\n");  return;  }  s->data[++(s->top)] = item;  }  int pop(Stack\* s) {  if (0 > s->top) {  printf("EMPTY\n");  return -1;  }  return s->data[(s->top)--];  }  void init(Stack\* s) {  s->top = -1;  }  int main() {  Stack pal1, pal2;  init(&pal1);  init(&pal2);  char str[101], r\_str[101];  printf("문자열을 입력하시오: ");  gets(str);  for (int i = 0;i < strlen(str);i++) {  int asc = str[i];  if ((65 <= asc && asc < 91) || (97 <= asc && asc < 123)) {  if (asc >= 65 && asc < 91) asc += 32;  r\_str[len++] = (char)asc;  }  }    // 스페이스, 대소문자, 구두점 등을 무시한 결과  //for (int i = 0;i < len;i++)  // printf("%c", r\_str[i]);  //printf("\n\n");  int s = 0, e = len - 1;  while (s <= e) {  push(&pal1, r\_str[s++]);  push(&pal2, r\_str[e--]);  }  while (pal1.top != -1 || pal2.top != -1) {  if (pop(&pal1) != pop(&pal2)) {  printf("회문이 아닙니다.\n");  exit(0);  }  }  printf("회문입니다.\n");  return 0;  }  실행 결과      문제의 내용대로 입력한 결과 |