|  |  |  |  |
| --- | --- | --- | --- |
| 2022/2 『자료구조』과제 보고서 | | | |
| 제목 | 3장 실습( ) 과제( O ) | 제출일자 | 2022.  10 .    05 . |
| 학번 | 201911608 | 이름 | 김지환 |

|  |
| --- |
| 02. float a[100]으로 선언된 배열의 시작 주소를 1000번지라고 할 때, 배열의 10번째 요소의 주소는 몇 번지인가?  (1) 1010번지 (2) 1020번지 (3) 1030번지 (4) 1040번지  -> float 은 4byte의 자료형이다. 4byte \* 10 = 40byte, 10번째 요소의 주소는 1040번지이다. |
| 04. 크기가 10인 배열 two[]를 선언하고 여기에 2의 제곱 값들을 저장해보자. 즉 배열의 첫 번째 요소에는 2^0을 저장하고 두 번째 요소에는 2^1값을 저장한다. 마지막 요소에는 2^9값을 저장한다. for 루프를 이용하여 배열의 전체 요소의 값을 출력하는 프로그램을 작성하라.  #include <stdio.h>  #define INDEX 10  int two[INDEX];  int main() {  for (int i = 0; i < INDEX;i++)  if (i == 0) two[i] = 1;  else two[i] = two[i - 1] \* 2;  for (int i = 0;i < INDEX - 1;i++)  printf("[%d]", two[i]);  printf("[%d]", two[INDEX - 1]);  return 0;  }  실행결과 |
| 06. C언어의 typedef를 이용하여 complex라고 하는 새로운 자료형을 정의하라. complex 자료형은 구조체로서 float형인 real변수와 역시 float형인 imaginary 변수를 갖는다. complex 자료형으로 변수 c1과 c2를 선언하여보아라.  #include <stdio.h>  #define INDEX 10  typedef struct {  float real;  float imaginary;  }complex;  int main() {  complex c1 = { 3.0, 2.0 };  complex c2 = { 2.0, 1.0 };  printf("c1 - 실수부 : %3.1f, 허수부 : %3.1fi\n", c1.real, c1.imaginary);  printf("c2 - 실수부 : %3.1f, 허수부 : %3.1fi\n", c2.real, c2.imaginary);  }  실행결과 |
| 08. 크기가 n인 배열 array에서 임의의 위치 loc에 정수 value를 삽입하는 함수 insert()를 작성하라. 정수가 삽입되면 그 뒤에 있는 정숟르은 한 칸씩 뒤로 밀려야 한다. 현재 배열에 들어있는 원소의 개수는 items개라고 하자. (여기서 items << n라고 가정)  void insert(int array[], int loc, int value) { .... }  #define \_CRT\_SECURE\_NO\_WARNINGS  #include <stdio.h>  #include <malloc.h>  int n, address, items = 0;  void insert(int array[], int loc, int value) {  printf("\n\ninsert, index : %d, value : %d", loc, value);  if (loc >= n) {  printf(" ====> #### error #### %d 번째 index는 없습니다.", loc);  return;  }  if (array[n - 1] != address) items--;  int temp = array[loc], temp1;  array[loc] = value;  for (int i = loc + 1; i < n; i++) {  temp1 = array[i];  array[i] = temp;  temp = temp1;  }  items++;  }  void print(int arr[]) {  printf("\n====== print arr ======\n");  for (int i = 0;i < n - 1;i++)  if (arr[i] != address)  printf("[%d]", arr[i]);  else  printf("[ ]");  if (arr[n - 1] != address)  printf("[%d]", arr[n - 1]);  else  printf("[ ]");  printf(" ---- items : %d\n", items);  }  int main() {  printf("배열 크기 입력(N) : ");  scanf("%d", &n);  int\* arr = malloc(sizeof(int) \* n);  address = arr[0];  printf("\n====== malloc arr ======\n\n");  for (int i = 0;i < n;i++)  printf("%d ", arr[i]);  insert(arr, 0, 10);  print(arr);  insert(arr, 5, 5);  print(arr);  insert(arr, 6, 3);  print(arr);  insert(arr, 5, 4);  print(arr);  insert(arr, 7, 5);  print(arr);  free(arr);  return 0;  }  실행결과 |
| 10. 크기가 n인 배열 array에서 임의의 위치 loc에 있는 정수를 삭제하는 함수 delete()를 작성하라. 정수가 삭제되면 그 뒤에 있는 정수들은 한 칸씩 앞으로 이동하여야 한다. 현재 배열에 들어 있는 원소의 개수는 items개라고 하자. (여기서 imtes << n 라고 가정)  int delete(int array[], int loc) { ... }  #define \_CRT\_SECURE\_NO\_WARNINGS  #include <stdio.h>  #include <malloc.h>  int n, address, items = 0;  int delete(int array[], int loc) {  printf("\n\n%d번 째 값 삭제\n", loc);  if (loc >= n) return 0;  if (array[loc] == address) return 0;  array[loc] = address;  items--;  int temp;  for (int i = loc; i < n-1; i++) {  temp = array[i+1];  array[i] = temp;  }  array[n - 1] = address;  return 1;  }  void insert(int array[], int loc, int value) {  printf("insert, index : %d, value : %d\n", loc, value);  if (loc >= n) {  printf(" ====> #### error #### %d 번째 index는 없습니다.\n\n", loc);  return;  }  if (array[n - 1] != address) items--;  int temp = array[loc], temp1;  array[loc] = value;  for (int i = loc + 1; i < n ; i++) {  temp1 = array[i];  array[i] = temp;  temp = temp1;  }  items++;  }  void print(int arr[]) {  printf("====== print arr ======\n");  for (int i = 0;i < n - 1;i++)  if (arr[i] != address)  printf("[%d]", arr[i]);  else  printf("[ ]");  if (arr[n-1] != address)  printf("[%d]", arr[n-1]);  else  printf("[ ]");  printf(" ---- items : %d\n",items);  }  int main() {  printf("배열 크기 입력(N) : ");  scanf("%d", &n);  int\* arr = malloc(sizeof(int) \* n);  address = arr[0];  printf("\n====== malloc arr ======\n\n");  for (int i = 0;i < n;i++)  printf("%d ", arr[i]);  printf("\n\n");  for (int i = 0; i < 5; i++) insert(arr, i \* 2, i + 1);  print(arr);  for (int i = 0; i < n; i++)  delete(arr, i) ? print(arr) : printf("index가 잘못되었거나 해당 index에 값이 존재하지 않습니다.\n");  free(arr);  return 0;  }  실행결과  -> 다음 페이지 |
| 12. 1개의 정수와 최대 크기가 20인 문자열로 이루어진 구조체를 저장할 수 있도록 동적 메모리를 할당 받고 여기에 정수 100과 문자열 “just testing”을 저장한 다음, 동적 메모리를 반납하는 프로그램을 작성하라.  #include <stdio.h>  #include <malloc.h>  typedef struct {  int n;  char s[21];  }store;  int main() {  store\* s;  s = (store\*)malloc(sizeof(store));  s->n = 100;  strcpy(s->s,"just testing");  printf("정수 값 : %d, 문자열 : %s", s->n, s->s);  free(s);  }  실행결과 |