컴프 자주 쓰이는 함수

1. 스트링 접합, 정렬

|  |
| --- |
| int mergeString(char c[], char a[], char b[]) { |
|  | int i, j; |
|  | int sizeA = 0, sizeB = 0, sizeC = 0; |
|  | char temp; |
|  | int minIndex; |
|  |  |
|  | for (i = 0; a[i] != '\0'; i++) sizeA = i; |
|  | for (i = 0; b[i] != '\0'; i++) sizeB = i; |
|  | sizeC = (sizeA + 1) + (sizeB + 1); |
|  |  |
|  | **//단순 합치기** |
|  | **for (i = 0; a[i] != '\0'; i++)** |
|  | **c[i] = a[i];** |
|  | **for (i = 0; b[i] != '\0'; i++)** |
|  | **c[sizeA + 1 + i] = b[i];** |
|  | **c[sizeC] = '\0';** |
|  |  |
|  | **//정렬하기** |
|  | **for (i = 0; i < sizeC - 1; i++) {** |
|  | **minIndex = i;** |
|  | **for (j = i + 1; j < sizeC; j++)** |
|  | **if (c[j] <= c[minIndex])** |
|  | **minIndex = j;** |
|  | **temp = c[i];** |
|  | **c[i] = c[minIndex];** |
|  | **c[minIndex] = temp;** |
|  | **}** |
|  |  |
|  | return 0; |
|  | } |

1. 선택정렬 함수

|  |
| --- |
| void selectionSort(int list[], int size) { |
|  | int i, j, temp, minIndex; |
|  |  |
|  | for (i = 0; i < size - 1; i++) { |
|  | minIndex = i; |
|  | for (j = i+1; j < size; j++) |
|  | if (list[j] < list[minIndex]) |
|  | minIndex = j; |
|  |  |
|  | temp = list[i]; |
|  | list[i] = list[minIndex]; |
|  | list[minIndex] = temp; |
|  | } |
|  | } |

1. 난수발생시키기

|  |  |
| --- | --- |
| #include<stdio.h> | |
|  | | #include <time.h> |
| void initArray(int arr[], int size) { |
|  | int i; | |
|  | for (i = 0; i < size; i++) | |
|  | arr[i] = rand() % 100; | |
|  | return; | |
|  | } | |

1. 재귀함수 예 (조합)

|  |
| --- |
| #define \_CRT\_SECURE\_NO\_WARNINGS |
|  | #include <stdio.h> |
|  |  |
|  |  |
|  |  |
|  | int comb(int n, int r) { |
|  | if (r == 0 || r == n) |
|  | return comb(n - 1, r - 1) + comb(n - 1, r); |
|  | } |
|  | int main(void) { |
|  | int n, r; |
|  |  |
|  | printf("Enter n and r:"); |
|  | scanf("%d %d", &n, &r); |
|  | printf("%d", comb(n, r)); |
|  | return 0; |
|  | } |

1. 전치행렬

|  |
| --- |
| int main(void) { |
|  | int arr1[2][4] = { 1,2,3,4,5,6,7,8 }; |
|  | int arr2[4][2]; |
|  | int arr3[8]; |
|  | int i, j; |
|  |  |
|  | /\*이동(복사) A -> B \*/ |
|  | for (i = 0; i < 4; i++) |
|  | for (j = 0; j < 2; j++) |
|  | arr2[i][j] = arr1[j][i]; |
|  |  |
|  | /\*이동(복사) A -> C \*/ |
|  | for (i = 0; i < 4; i++) |
|  | arr3[i] = arr1[0][i]; |
|  | for (i = 4; i < 8; i++) |
|  | arr3[i] = arr1[1][i - 4]; |
|  |  |
|  | **/\*A,B,C(2차원배열) 출력\*/** |
|  | **printf("--배열 A \n");** |
|  | **for (i = 0; i < 2; i++)** |
|  | **for (j = 0; j < 4; j++) {** |
|  | **if (j == 3)** |
|  | **printf("%2d\n", arr1[i][j]);** |
|  | **else** |
|  | **printf("%2d", arr1[i][j]);** |
|  | **}** |
|  |  |
|  |  |
|  | printf("--배열 B \n"); |
|  | for(i=0;i<4;i++) |
|  | for (j = 0; j < 2; j++) { |
|  | if (j == 1) |
|  | printf("%2d\n", arr2[i][j]); |
|  | else |
|  | printf("%2d", arr2[i][j]); |
|  | } |
|  |  |
|  | printf("--배열 C\n"); |
|  | for (i = 0; i < 8; i++) |
|  | printf("%2d", arr3[i]); |
|  | printf("\n"); |
|  | } |