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"); |
|  | } |

1. Row 최대값
2. for (i = 0; i < 4; i++) {
3. max = test[i][0];
4. for (j = 0; j < 3; j++) {
5. if (max <= test[i][j])
6. max = test[i][j];
7. }
8. sumMax += max;
9. printf("%d번째 row에서의 최대값: %d\n", i + 1, max);
10. }

7. 전치행렬

For(i=0;i<2;i++)

For(j=0lj<4;j++)

Arr3[i\*4+j]=arr1[i][j];

8. 집합관련

int setUnion(int set1[], int size1, int set2[], int size2, int setResult[])

{

// 구현 필요 합집합

int i, j, setResultSize;

for (i = 0; i < size1; i++)

setResult[i] = set1[i];

setResultSize = i;

for (j = 0; j < size2; j++) {

if (!hasElement(setResult, setResultSize, set2[j])) { //원소가 없을 때

setResult[setResultSize] = set2[j];

setResultSize++;

}

}

return setResultSize;

}

int setIntersecton(int set1[], int size1, int set2[], int size2, int setResult[])

{

// 구현 필요 교집합

int i = 0;

int j;

for (j = 0; j < size2; j++) {

if (hasElement(set1, size1, set2[i])) //공통 원소가 있음

setResult[i] = set2[j];

i++;

}

return i;

}

int setComplements(int set1[], int size1, int set2[], int size2, int setResult[])

{

// 구현 필요 차집합

int i = 0, j;

for (j = 0; j < size1; j++)

if (!hasElement(set2, size2, set1[j])) //공통없음

setResult[i++] = set1[j];

return i;

}

9. 역순배열 만들기

for (i = size - 1; i >= 0; i--)

str2[(size - 1) - i] = str[i];

str2[size] = '\0';

10.문자열길이

For(size=0;s[size]!=’\0’;size++);

11. 소수 판별 함수

Int isPrime(int num){

Int i;

For(i=2;i<num;i++)

If(num%i==0) return 0;

Return 1

}

12. 값 찾기(char 타입)

int serchIndex(char s[], char key) {

for (i = 0; s[i] != '\0'; i++)

if (s[i] == key)

return i + 1;

return -1;

}