2022 CAT-CERT 스터디 보고서

컴퓨터정보공학부 202221168 김지혜

2022.05.04 #C\_study-5

포인터 실습

소스코드

#include <stdio.h>

void swap(int\* a, int \*b) {

int tmp = \*b;

\*b = \*a;

\*a = tmp;

}

int main() {

int a = 1, b = 7;

printf("%d %d\n", a, b);

swap(&a, &b);

printf("%d %d\n", a, b);

}

출력 결과

텍스트이(가) 표시된 사진

자동 생성된 설명

포인터 실습2

소스코드

#include<stdio.h>

void reverse(char\* a, int b);

int main() {

int arr[5] = { 1,2,3,4,5 };

printf("BEFORE :");

for (int i = 0; i < 5; i++)

printf("%d ", arr[i]);

reverse(arr, 5);

printf("\nAFTER : ");

for (int i = 0; i < 5; i++)

printf("%d ", \*(arr + i));

printf("\n");

}

void reverse(int\* a, int b) {

int tmp = \*a;

\*a = b;

a[b - 1] = tmp;

}

출력결과

텍스트이(가) 표시된 사진

자동 생성된 설명

포인터 실습3

소스코드

#include<stdio.h>

void sort(arr);

int main() {

int arr[100];

for (int i = 0; i < 10; i++)

scanf("%d", &arr[i]);

sort(arr);

for (int i = 0; i < 10; i++)

printf("%d ", arr[i]);

}

void sort(int\* a) {

int temp = 0;

for (int i = 0; i < 10 - 1; i++) {

for (int j = 0; j < 10 - 1 - i; j++){

if (a[j] < a[j + 1]) {

temp= a[j];

a[j]=a[j+1];

a[j+1]=temp;

}

}

}

}

출력결과

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

#실습과제 1

소스코드

#include<stdio.h>

int main() {

int score, tmp;

int sum = 0;

int scorelist[9] = { 0, };

int numberlist[9] = { 0, };

int\* ptr = &scorelist;

for (int i = 0; i < 8; i++) {

scanf("%d", &score);

scorelist[i] = score;

numberlist[i] = score;

}

for (int j = 0; j < 7; j++) {

for (int k = 0; k < 7 - j; k++) {

if (scorelist[k] < scorelist[k + 1]) {

tmp = scorelist[k];

scorelist[k] = scorelist[k + 1];

scorelist[k + 1] = tmp;

}

}

}

for (int i = 0; i < 5; i++) {

sum = sum + scorelist[i];

}

printf("%d\n", sum);

for (int j = 0; j < 8; j++) {

for (int i = 0; i < 5; i++) {

if (numberlist[j] == scorelist[i]) {

printf("%d ", j + 1);

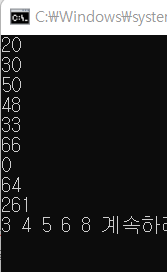
}

}

}

}

출력결과



#실습과제 2

소스코드

#include<stdio.h>

#include<string.h>

int main() {

int a;

char arr1[100];

char arr2[100];

int count1[26] = { 0, };

int count2[26] = { 0, };

int tmp = 0;

int tmpp = 0;

int answer = 0;

scanf("%d", &a);

for (int i = 0; i < a; i++) {

scanf("%s %s", arr1, arr2);

int len1 = strlen(arr1);

int len2 = strlen(arr2);

if (len1 != len2) {

printf("%s & %s are NOT anagrams.\n", arr1, arr2);

}

else {

for (int i = 0; i < len1; i++) {

tmp = (int)arr1[i];

count1[tmp - 97]++;

}

for (int j = 0; j < len2; j++) {

tmpp = (int)arr2[j];

count2[tmpp - 97]++;

}

for (int i = 0; i < 26; i++) {

if (count1[i] == count2[i]) {

answer++;

}

}

if (answer == 26) {

printf("%s & %s are anagrams.\n", arr1, arr2);

}

else {

printf("%s & %s are NOT anagrams.\n", arr1, arr2);

}

answer = 0;

}

}

}

출력결과

텍스트이(가) 표시된 사진

자동 생성된 설명

#실습과제 3

소스코드

#include<stdio.h>

int main() {

char arr[100000];

scanf("%s", arr);

int count1 = 0;

int count2 = 0;

for (int i = 0; arr[i] != '\0'; i++) {

if (arr[i] == '(')

count1++;

else if (arr[i] == ')')

count2++;

}

if (count1 == count2) {

printf("True\n");

}

else if (count1 != count2) {

printf("false\n");

}

}

출력결과

텍스트이(가) 표시된 사진

자동 생성된 설명

#실습과제 4

소스코드

#include<stdio.h>

int main() {

int num;

int i;

int j = 0;

int p = 0;

int arr[100] = { 0, };

int odd[100] = { 0, };

int even[100] = { 0, };

int tmp = 0;

int tmpp=0;

scanf("%d", &num);

for (i = 0; i < num; i++) {

scanf("%d", &arr[i]);

if ((arr[i] % 2)== 0) {

even[p] = arr[i];

p++;

}

else if((arr[i]%2)!=0) {

odd[j] = arr[i];

j++;

}

}

for (int i = 0; i < num/2; i++) {

for (int j = 0; j < (num/2)-1; j++) {

if (odd[j] >odd[j + 1]) {

tmp = odd[j];

odd[j] = odd[j + 1];

odd[j + 1] = tmp;

}

}

}

for (int i = 0; i < num/2; i++) {

for (int j = 0; j < (num/2)-1; j++) {

if (even[j] > even[j + 1]) {

tmpp = even[j];

even[j] = even[j + 1];

even[j + 1] = tmpp;

}

}

}

for (int i = 0; i < num/2; i++) {

printf("%d ", odd[i]);

}

for (int i = 0; i < num/2; i++) {

printf("%d ", even[i]);

}

}

출력과제

텍스트이(가) 표시된 사진

자동 생성된 설명

#실습과제 5

소스코드

#include<stdio.h>

#include<string.h>

int main() {

int count = 0;

char arr[100] = { 0, };

int length = 0;

scanf("%[^\n]s", arr);

length = strlen(arr);

for (int i = 0; i<length; i++) {

if (arr[i] == ' ') {

count++;

}

}

printf("%d\n", count );

}

출력결과

텍스트이(가) 표시된 사진

자동 생성된 설명

시

#1 math.h에 대해 조사한 후, math.h에 포함된 라이브러리 함수 조사 및 실습

Math.h 헤더파일은 수학과 관련된 수학연산을 위한 정보를 포함함

\*C언어는 함수 오버로딩을 지원하지 않아서 double 타입으로만 존재

|  |  |
| --- | --- |
| 1.double sqrt( double x ) | 루트 x(x제곱근)을 반환함 |
| #include<stdio.h>  #include<math.h>  int main() {  double result;  result = sqrt(9.0);  printf("%lf\n", result);  } |  |

|  |  |
| --- | --- |
| 2.double pow( double x, double y ) | x의 y제곱의 값을 반환함 |
| #include<stdio.h>  #include<math.h>  int main() {  double result;  result = pow(9.0,2.0);  printf("%lf\n", result);  } |  |

|  |  |
| --- | --- |
| 3.double fabs( double x )  double abs( double x ) | x의 절대값을 반환하는 함수입니다 |
| #include<stdio.h>  #include<math.h>  int main() {  double result1;  double result2;  result1 =fabs(9.0);  result2 = fabs(-9, 0);  printf("%lf\n", result1);  printf("%lf\n", result2);  } |  |

|  |  |
| --- | --- |
| 4. double ceil( double x ) | x보다 작지 않은 최소 크기의 정수를 반환함  (올림함수) |
| #include<stdio.h>  #include<math.h>  int main() {  double result;  result =ceil(3.2);  printf("%lf\n", result);  } |  |

|  |  |
| --- | --- |
| 5. double floor( double x ) | x보다 크지 않은 최대크기의 정수를 반환함 |
| #include<stdio.h>  #include<math.h>  int main() {  double result;  result =floor(3.2);  printf("%lf\n", result);  } |  |

|  |  |
| --- | --- |
| 6.double exp( double x ) | 무리수 e의 x제곱을 반환함 |
| #include<stdio.h>  #include<math.h>  int main() {  double result;  result =exp(3.0);  printf("%lf\n", result);  } |  |

|  |  |
| --- | --- |
| 7. double log( double x )  double log10( double x )  double log2( double x ) | 자연로그 x의 값을 반환함  상용로그(밑이 10) x의 값을 반환함  밑이 2인 로그 x의 값을 반환함 |
| #include<stdio.h>  #include<math.h>  int main() {  double result1;  double result2;  double result3;  result1 =log(10.0);  result2 = log10(10.0);  result3 = log2(10.0);  printf("%lf\n", result1);  printf("%lf\n", result2);  printf("%lf\n", result3);  } |  |

|  |  |
| --- | --- |
| 8.double cos( double x )  double sin( double x )  double tan( double x )  \*각도는 라디안으로 처리 | x의 코사인(cos) 값을 반환함  x의 사인(sin) 값을 반환함  x의 탄젠트(tan) 값을 반환함 |
| #include<stdio.h>  #include<math.h>  int main() {  double pi = 3.141592;  printf("%.lf\n", cos(pi));  printf("%.lf\n", sin(pi));  printf("%.lf\n", tan(pi));  } |  |

#2 32bit vs 64bit 차이 자세하게 조사

: 저장 장치의 bit폭 차이

cpu가정보를처리하는 최소단위를 레지스터(임시기억장치)

매번 메모리에서 cpu로 데이터를 가져오면 지연이 발생하기 때문에 cpu에서 한번에 많이 처리할수록 좋아지는데 그 처리할 수 있는 양의 한계치가 32비트 또는 64비트

Bit - cpu가 정보를 처리하는 방식

-0과 1로 기호를 쓰며 두 수로 이루어짐(이진법)

-2비트는 2의 32제곱, 64비트는 2의 64제곱

한 번에 취급할 수 있는 정보가 증가한다는 것은 많은 양의 데이터를 처리하게 되므로 처리의 고속화도 도모할 수 있게 됨, 발전과 함께 bit 수가 증가하면 기능이 고도화됨

* 32비트보다 64비트가 기능이 더 좋음
* 보통 64비트 cpu의 경우 32비트도 지원하지만 32비트 cpu에 64비트 os를 적용시키면 성능이 떨어짐
* 32비트는 x86 64비트는 x64이다(PC 칩셋의 인텔 품번으로 인해 86)

저장장치의 용량 단위가 점차 커지면서 기존에 있던 장치들과 새로 생긴 장치들 사이의 호환문제가 발생하게 됨. 똑같은 프로그램일지라도 32비트용과 64비트용이 나눠짐