**포인터 실습**

|  |
| --- |
| #define \_CRT\_SECURE\_NO\_WARNINGS  #include<stdio.h>  void swap(int\* a, int\* b) {  int tmp = 0;  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);  } |

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**포인터 실습2**

|  |
| --- |
| #define \_CRT\_SECURE\_NO\_WARNINGS  #include<stdio.h>  void reverse(int arr[], int a ) {    int rev[5];  for (int i = 0; i < a; i++)  rev[i] = arr[4 - i];  for (int i = 0; i < a; i++)  \*(arr + i) = \*(rev + i);    }  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");  } |

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**포인터 실습3**

|  |
| --- |
| #define \_CRT\_SECURE\_NO\_WARNINGS  #include<stdio.h>  void sort(int arr[]) {  for (int i = 0; i < 10; i++) {  for (int j = 0; j < 9; j++) {  if (arr[j] < arr[j + 1]) {  int tmp = 0;  tmp = \*(arr+j+1);  arr[j + 1] = \*(arr + j);  arr[j] = tmp;  }  }  }  }  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]);  } |

/\*int score = 0;

score = (\*(arr)+\*(arr + 1) + \*(arr + 2) + \*(arr + 3) + \*(arr + 4)) ;

printf(score);

\*/

**실습 과제 1 시험**

|  |
| --- |
| #define \_CRT\_SECURE\_NO\_WARNINGS  #include<stdio.h>  void sort(int arr[]) {  for (int i = 0; i < 8; i++) {  for (int j = 0; j < 7; j++) {  if (arr[j] < arr[j + 1]) {  int tmp = 0;  tmp = \*(arr + j + 1);  arr[j + 1] = \*(arr + j);  arr[j] = tmp;  }  }  }  }  int main() {  int arr[10];  int sub[10];  int num[10] = { 0 };  for (int i = 0; i < 8; i++)  scanf("%d", &arr[i]);  for (int i = 0; i < 8; i++)  sub[i] = arr[i];  sort(arr);  int score = 0;  score = \*(arr) + \*(arr + 1) + \*(arr + 2) + \*(arr + 3) + \*(arr + 4) ;  printf("%d\n", score);    for (int i = 0; i < 5; i++) {  for (int z = 0; z < 9; z++) {  if (sub[z] == arr[i])  num[i] = z + 1;    }  }  sort(num);  for (int i = 4; i >= 0; i--)  printf("%d ", num[i]);  } |

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**실습#2 애너그램**

|  |
| --- |
| #define \_CRT\_SECURE\_NO\_WARNINGS  #include<stdio.h>  int main() {  int num = 0;  scanf("%d", &num);  int count = 0;  char arr1[100][100];  char arr2[100][100];    for (int i = 0; i < num; i++)  scanf("%s %s", &arr1[i], &arr2[i]);    for (int i = 0; i < num; i++) {  for (int z = 0; z < strlen(arr1[i]); z++) {  if (arr1[i][z] != arr2[i][z]){  for (int j = 0; j < strlen(arr2[i]); j++)  if (arr1[i][j] == arr2[i][z]) {  count++;  break;  }  }  else if (arr1[i][z] == arr2[i][z]) {  for (int j = 0; j < strlen(arr2[i]); j++)  if (arr1[i][j] == arr2[i][z]) {  count++;  break;  }  }  }  if (count == strlen(arr2[i])) {  printf("%s & %s are anagrams\n", arr1[i], arr2[i]);  count = 0;  }  else {  printf("%s & %s are NOT anagrams\n", arr1[i], arr2[i]);  count = 0;  }  }    } |

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**실습 #3 괄호**

|  |
| --- |
| #define \_CRT\_SECURE\_NO\_WARNINGS  #include<stdio.h>  int main() {  char arr[100000];  scanf("%s", arr);    int lef = 0;  int rig = 0;  if (arr[0] != '(')  printf("false");  for (int i = 0; i < strlen(arr); i++) {  if (arr[i] == '(')  lef++;  else if (arr[i] == ')')  rig++;  }  if (lef != rig)  printf("false");  if (lef == rig)  printf("true");  } |

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**실습 #4 강박증**

|  |
| --- |
| #define \_CRT\_SECURE\_NO\_WARNINGS  #include<stdio.h>  int num;  void sort(int arr[]) {  for (int i = 0; i <num; i++) {  for (int j = 0; j < num-1; j++) {  if (arr[j] < arr[j + 1]) {  int tmp = 0;  tmp = \*(arr + j+1);  arr[j +1] = \*(arr + j);  arr[j] = tmp;  }  }  }  }  int main() {  scanf("%d", &num);  int arr[100];  int num1[50] = { 0 };  int num2[50] = { 0 };  int count1 = 1;  int count2 = 1;  for (int i = 0; i < num; i++)  scanf("%d", &arr[i]);  for (int i = 0; i < num; i++) {  if (arr[i] % 2 == 1) {  num1[count1] = arr[i];  count1++;  }  if (arr[i] % 2 == 0) {  num2[count2] = arr[i];  count2++;  }  }  sort(num1);  sort(num2);  for (int i = num; i >=0 ; i--)  if(num1[i]!=0)  printf("%d ", num1[i]);  for (int i = num; i >= 0; i--)  if (num2[i] != 0)  printf("%d ", num2[i]);  } |

텍스트이(가) 표시된 사진

자동 생성된 설명

**실습 #5 단어의 개수**

|  |
| --- |
| #define \_CRT\_SECURE\_NO\_WARNINGS  #include<stdio.h>  int main() {  char arr[1000000];  scanf("%[^\n]s", arr);  int count = 0;  for (int i = 0; i < strlen(arr); i++) {  if (arr[i] == ' ')  count++;  }  printf("%d", count);    } |

**텍스트이(가) 표시된 사진

자동 생성된 설명**

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

**Math.h :** 여러 수학 함수들을 포함하는 C언어의 표준 라이브러리 / 대부분의 함수들이 부동소수점을 다루며 각도는 라디안을 사용함

#include <math.h> 사용

기본함수 21개, 상수\_변수함수 14개, 특수함수 7개 (총 42개)

**#sqrt 함수** 텍스트, 손목시계, 게이지이(가) 표시된 사진

자동 생성된 설명

|  |
| --- |
| #include<stdio.h>  #include<math.h>  int main() {  double num;  num = sqrt(16.0);  printf("%lf\n", num);  return 0;  } |

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**#pow 함수 – 거듭제곱함수 pow(float형 변수, float형 변수)**

|  |
| --- |
| #include<stdio.h>  #include<math.h>  int main() {    float num= pow(4 , 5);  printf("%lf\n", num);  } |

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**#ceil 함수 – x보다 작지 않은 가장 작은 정수를 구함**

|  |
| --- |
| #include<stdio.h>  #include<math.h>  int main() {  double num;  num = ceil (15.3);  printf("%lf\n", num);  return 0;  } |

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**#abs 함수 – 절댓값**

|  |
| --- |
| #include<stdio.h>  #include<math.h>  int main() {  double num;  num = abs (-3);  printf("%lf\n", num);  return 0;  } |

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**#exp 함수 - 를 구함**

|  |
| --- |
| #include<stdio.h>  #include<math.h>  int main() {  double num;  num = exp (3);  printf("%lf\n", num);  return 0;  } |

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**Fmod 함수 – x를 y로 나눈 나머지 구함**

|  |
| --- |
| #include<stdio.h>  #include<math.h>  int main() {  double num;  num = fmod (5,3);  printf("%lf\n", num);  return 0;  } |

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**+ double 형과 float 차이**

Float – 소수점 이하 6자리

Double – 소수점 이하 15자리

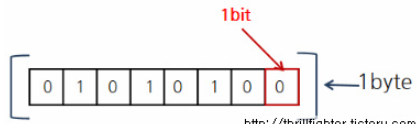
**#2 32bit vs 64bit 차이 자세하게 조사**

**Q. bit란?**

**A.** 컴퓨터의 **처리 정보의 최소 단위** “binary digit” / binary – 2진수, digit – 자리수를 의미함

컴퓨터는 모든 데이터를 이진법으로 처리 /

1bit는 0 또는 1의 상태 유지 (그 의외의 값 X) / 1바이트는 8bit



CPU가 처리하는 데이터의 최소단위 (레지스터)가 몇 bit 인지? → 32 bit / 64 bit

|  |  |  |
| --- | --- | --- |
|  | **32bit** | **64bit** |
| **CPU** | 32bitCPU | 64bitCPU |
| **레지스터** | (2의 32승)개 저장 가능 | (2의 64승)개 저장 가능 |
| **메모리** | 최대 4GB | 16EB |
| **속도** |  | 더 빠름 |

64bit의 CPU는 32bit 명령 실행 가능 ( 64 bit 시스템은 32 bit 시스템에서 사용되는 모든 명령어를 가지고 있음)

→ 32bit의 CPU는 64bit 명령 수행 X

**※** Linux의 경우 LFS와 Gentoo 같은 소스 기반 배포판에서 32bit 바이너리를 뺀 64 bit 시스템 구축 가능

**√ 64 bit 속도가 더 빠른 이유**

컴퓨터는 낮은 메모리를 보완하기 위해 일부 데이터를 느린 하드 드라이브에 보관

64bit는 더 빠른 RAM에 저장한 후, 느린 하드 드라이브에 저장하고 더 많은 데이터를 RAM에 저장할 수 있음 → 컴퓨터가 빠르게 실행 가능

**√ 메모리**

3D 응용 프로그램, 동영상 편집, 과학 계산 → 메모리 크게 소모

32 bit 윈도우에서 8GB 시스템을 돌릴 수 있어도 실제론 4GB 만 인식함

64bit 가 32bit에 비해 정밀한 그래픽 처리 가능

**√ 윈도우 정보**

X86 = 32 bit (80-86이라는 숫자를 인텔 32 bit 이하 계열 제품명에 붙였기 때문에 x86으로 표현)

X64 = 64 bit