2023 학년도 1학기 출석과제물

|  |  |  |  |
| --- | --- | --- | --- |
|  | **교과목명** | : | C프로그래밍 |
|  | **학번** | : | 202234-366307 |
|  | **성명** | : | 최문성 |
|  | **연락처** | : | 010-8010-6050 |

1. 예제 1-1
   1. 소스코드

*/\* 화면에 Hello~를 출력하는 프로그램*

*파일 이름 : exam1-1.c*

*수정 일자 : 2023-04-17*

*작성자 : 최문성*

*학번 : 202234-366307*

*\*/*

*#include* <stdio.h> *// 스탠다드 입출력 헤더파일 추가하기*

int main() *// main 함수*

{ *// 함수 여는 중괄호*

printf("Hello~ \n"); *// 화면에 Hello~ 출력*

} *// 함수 닫는 중괄호, main 함수 끝*

* 1. 실행결과

A screenshot of a computer

Description automatically generated with medium confidence

1. 예제 1-2
   1. 소스코드

*/\* 사용자로부터 정수 두 개를 입력받아 곱한 결과를 출력하는 프로그램 \*/*

*#include* <stdio.h> *// 표준 입출력 헤더파일 추가*

*#pragma* warning(disable:4996) *// 보안 경고를 무시*

int a, b, c; *// 사용할 변수 정수로 선언*

int product(int *x*, int *y*); *// 함수 원형 선언*

void main() *// 메인 함수 시작*

{

    printf("Enter a number between 1 and 100: "); *// 사용자에게 메세지 출력*

    scanf("%d", &a); *// 첫번째 수를 입력받고 값을 a에 저장*

    printf("Enter another number between 1 and 100: "); *// 사용자에게 메세지 출력*

    scanf("%d", &b); *// 두번째 수를 입력받고 값을 b에 저장*

    c = product(a, b); *// product() 함수 호출하여 c에 저장*

    printf("%d \* %d = %d \n", a, b, c); *// 저장한 결과 수식 형태로 출력*

} *// 메인 함수 끝*

int product(int *x*, int *y*) *// 두 수의 곱을 구하고 반환하는 함수 정의*

{

*return* *x* \* *y*;

}

* 1. 실행결과

Text

Description automatically generated

1. 예제 2-1
   1. 소스코드

*// c언어의 자료형에 대한 크기를 나타내는 프로그램*

*#include* <stdio.h> *// 표준 입출력 헤더파일 추가*

void main() *// 메인 함수 시작*

{ *// 메인 함수 여는 중괄호*

    printf("char 자료형 크기 : %d byte byte\n", sizeof(char)); *// char 자료형의 크기 출력*

    printf("signed char 자료형 크기 : %d byte\n", sizeof(signed char)); *// signed char 자료형의 크기 출력*

    printf("unsigned char 자료형 크기 : %d byte\n", sizeof(unsigned char)); *// unsigned char 자료형의 크기 출력*

    printf("short 자료형 크기 : %d byte\n", sizeof(short)); *// short 자료형의 크기 출력*

    printf("short int 자료형 크기 : %d byte\n", sizeof(short int)); *// short int 자료형의 크기 출력*

    printf("signed short 자료형 크기 : %d byte\n", sizeof(signed short)); *// signed short 자료형의 크기 출력*

    printf("signed short int 자료형 크기 : %d byte\n", sizeof(signed short int)); *// signed short int 자료형의 크기 출력*

    printf("unsigned short 자료형 크기 : %d byte\n", sizeof(unsigned short)); *// unsigned short 자료형의 크기 출력*

    printf("unsigned short int 자료형 크기 : %d byte\n", sizeof(unsigned short int)); *// unsigned short int 자료형의 크기 출력*

    printf("int 자료형 크기 : %d byte\n", sizeof(int)); *// int 자료형의 크기 출력*

    printf("signed int 자료형 크기 : %d byte\n", sizeof(signed int)); *// signed int 자료형의 크기 출력*

    printf("unsigned 자료형 크기 : %d byte\n", sizeof(unsigned)); *// unsigned 자료형의 크기 출력*

    printf("unsigned int 자료형 크기 : %d byte\n", sizeof(unsigned int)); *// unsigned int 자료형의 크기 출력*

    printf("long 자료형 크기 : %d byte\n", sizeof(long)); *// long 자료형의 크기 출력*

    printf("long int 자료형 크기 : %d byte\n", sizeof(long int)); *// long int 자료형의 크기 출력*

    printf("signed long 자료형 크기 : %d byte\n", sizeof(signed long)); *// signed long 자료형의 크기 출력*

    printf("signed long int 자료형 크기 : %d byte\n", sizeof(signed long int)); *// signed long int 자료형의 크기 출력*

    printf("unsigned long 자료형 크기 : %d byte\n", sizeof(unsigned long)); *// unsigned long 자료형의 크기 출력*

    printf("unsigned long int 자료형 크기 : %d byte\n", sizeof(unsigned long int)); *// unsigned long int 자료형의 크기 출력*

    printf("long long 자료형 크기 : %d byte\n", sizeof(long long)); *// long long 자료형의 크기 출력*

    printf("long long int 자료형 크기 : %d byte\n", sizeof(long long int)); *// long long int 자료형의 크기 출력*

    printf("signed long long 자료형 크기 : %d byte\n", sizeof(signed long long)); *// signed long long 자료형의 크기 출력*

    printf("signed long long int 자료형 크기 : %d byte\n", sizeof(signed long long int)); *// signed long long int 자료형의 크기 출력*

    printf("unsigned long long 자료형 크기 : %d byte\n", sizeof(unsigned long long)); *// unsigned long long 자료형의 크기 출력*

    printf("unsigned long long int 자료형 크기 : %d byte\n", sizeof(unsigned long long int)); *// unsigned long long int 자료형의 크기 출력*

    printf("float 자료형 크기 : %d byte\n", sizeof(float)); *// float 자료형의 크기 출력*

    printf("double 자료형 크기 : %d byte\n", sizeof(double)); *// double 자료형의 크기 출력*

    printf("long double 자료형 크기 : %d byte\n", sizeof(long double)); *// long double 자료형의 크기 출력*

} *// 메인 함수 끝*

* 1. 실행결과

Text

Description automatically generated

1. 예제 2-2
   1. 소스코드

*// #define 문을 이용한 상수 정의 예제*

*#include* <stdio.h> *// 표준 입출력 헤더파일 추가*

*#define* IN "원의 반지름은 100입니다. \n" *// 시작 메세지 정의*

*#define* PI 3.1415926353 *// 원주율 파이값 정의*

*#define* AREA(*X*) (PI \* (X) \* (X)) *// 원의 면적을 구하는 매크로 함수 정의*

*#define* OUT printf("원의 면적을 구하고 프로그램을 종료합니다. \n") *// 종료 메세지 출력 함수*

void main() *// 메인 함수 시작*

{

    printf("%s \n", IN); *// 시작 메세지 출력*

    printf("원의 면적은 %10.5f입니다. \n\n", AREA(100)); *// 원의 면적을 구하고 출력*

    OUT; *// 종료 메세지 출력*

}

* 1. 실행결과

Text

Description automatically generated

1. 예제 2-3
   1. exam2-3.h 소스코드

*#include* <stdio.h> *// 표준 입출력 헤더파일 추가*

*#define* ADD(*x*, *y*) ((x) + (y)) *// 매크로 함수 더하기 정의*

*#define* SUB(*x*, *y*) ((x) - (y)) *// 매크로 함수 빼기 정의*

* 1. exam2-3.c 소스코드

*#include* "exam2-3.h" *// exam2-3.h 헤더파일 추가*

*/\* exam2-3.h 에 이미 스탠다드 인풋 아웃풋 파일이 포함되어 있으므로 다시 불러오지 않아도 됨 \*/*

void main() *// 메인 함수 시작*

{

    int add\_result, sub\_result; *// 사용할 변수 정수로 선언*

    printf("10과 5를 더하면 %d이다. \n", ADD(10, 5)); *// ADD() 매크로 함수 호출하여 결과 출력*

    printf("10과 5를 빼면 %d이다. \n", SUB(10, 5)); *// SUB() 매크로 함수 호출하여 결과 출력*

}

* 1. 실행결과

Text

Description automatically generated