|  |  |  |  |
| --- | --- | --- | --- |
| 프로그래밍 실습 2 |  |  |  |
|  | | | |
| Report 1 | | | |
| 프로그래밍 실습 1 복습 | | | |
|  | | | |
|  |  | **학번** | 2018212236 |
|  |  | **학부** | 전자정보통신공학 |
|  |  | **이름** | 김동주 |
|  |  |  |  |
|  |  | **제출일자** | 2018-09-09 (수정 전)  2018-09-16 (수정 후) |
|  |  | **담당교수** | 반상우 |
|  |  |  |  |
|  |  | **작업환경** | Visual Studio Code |
|  |  | **컴파일러** | MinGW GCC |
|  |  |  |  |

# 

|  |  |  |  |
| --- | --- | --- | --- |
| **문제 내용** | 1학기 프로그래밍실습 1 내용 복습  - 각 단원별로 최소 1개의 프로그램 복습  - 총 10개 이상의 프로그램 복습  \* 프로그램 선택은 자율 : eclass에 등록된 1학기 복습자료 또는 교재 내용 활용하여 각자에게 가장 도움이 될 수 있는 프로그램 선택. | | |

## 1. [Ch. 2] C프로그래밍 첫걸음

|  |  |  |  |
| --- | --- | --- | --- |
| **내용** | Hello, world! - [p.74] putstring.c의 변형 | | |
| **코드** | #include <stdio.h>  int main(void) {  puts("Hello, world!\n");  return 0;  } | | |
| **실행결과** |  | | |
| **고찰** | 모든 프로그래머라면 한 번씩은 출력해 보았을 “Hello, world!” 를 출력하는 프로그램을 만들어보았다.  다시 새로 C언어를 시작하는 느낌이다. | | |

## 2. [Ch. 3] 자료형과 변수

|  |  |  |  |
| --- | --- | --- | --- |
| **내용** | 다음 조건을 만족하는 프로그램을 작성하시오.   * 문자형 연산 ‘A’ + 2 결과를 문자로 출력 * 문자형 연산 ‘A’ + 5 결과를 문자로 출력 * 문자형 연산 ‘S’ - 1 결과를 문자로 출력 * 문자형 연산 ‘S’ - 3 결과를 문자로 출력   [p.164] 18번 | | |
| **코드** | #include <stdio.h>  int main()  {  printf("'A' + 2 = %c\n", 'A' + 2);  printf("'A' + 5 = %c\n", 'A' + 5);  printf("'S' - 1 = %c\n", 'S' - 1);  printf("'S' - 3 = %c\n", 'S' - 3);  return 0;  } | | |
| **실행결과** |  | | |
| **고찰** | 교과서 164쪽의 18번 문제의 복습을 하였다.  문자형 변수에도 정수의 덧뺄셈을 할 수 있음을 다시금 떠올렸다. | | |

## 3. [Ch. 4] 전처리와 입출력

|  |  |  |  |
| --- | --- | --- | --- |
| **내용** | [p.171] macro.c | | |
| **코드** | #include <stdio.h>  #define KPOP 50000000  #define PI 3.14  #define PRT printf("매크로 상수 예제 종료\n");  int main(void) {  double radius = 2.83;  printf("한국인구: %d명\n", KPOP);  printf("원 면적: %f\n", radius \* radius \* PI);  PRT;  return 0;  } | | |
| **실행결과** |  | | |
| **고찰** | #define 전 처리기를 이용한 상수 사용은 1학기 활동을 통해 익숙해져 있었다. 하지만 함수를 전처리기를 통해 선언하는 것에 익숙하지 않았기에, 5번 줄의   |  | | --- | | #define PRT printf("매크로 상수 예제 종료\n"); |     을 보고 나서 자주 사용되는 함수인   |  | | --- | | puts("");  printf("\n"); |     와 같은 함수들을 전처리기를 이용하여 앞으로 사용해보고 싶다고 생각했다. | | |

## 4. [Ch. 4] 전처리와 입출력 (2)

|  |  |  |  |
| --- | --- | --- | --- |
| **내용** | [p.173] advancemacro.c | | |
| **코드** | #include <stdio.h>  #define MESSAGE "프로그램언어의 학습은 일반언어의 학습과 \  같이 반복학습이 중요하다"    #define PI 3.14  #define VOLUME(r) (4 \* PI \* CUBE(r) / 3)  #define SQUARE(x) ( (x) \* (x) )  #define CUBE(x) ( SQUARE(x) \* (x) )  #define MULT(x, y) ( (x) \* (y) )  int main()  {  double radius = 2.32;  printf("반지름이 %.2lf인 구의 체적은 %.2lf 입니다.", radius, VOLUME(radius));  printf("실수 %.2lf의 제곱은 %.2lf 입니다.\n", 4.29, SQUARE(4.29));  printf("실수 %.2lf의 제곱은 %.2lf 입니다.\n", 3.0, CUBE(3.0));  printf("실수 %.2lf와 실수 %.2lf의 곱은 %.2lf입니다.\n", 2.78, 3.62, MULT(2.78, 3.62));  puts(MESSAGE);  return 0;  } | | |
| **실행결과** |  | | |
| **고찰** | 앞선 4번 문제와 더불어 전처리기를 이용하는 방법의 심화 문제이다.  이번 문제에서는 매개변수를 사용하는 전처리기 함수가 있으며, 이들이 선언 될 때에는 중괄호 블럭 ‘{, }’ 이 아닌 일반 괄호 ‘(, )’를 사용한다.  잘 기억해두자. | | |

## 5. [Ch. 5] 연산자

|  |  |  |  |
| --- | --- | --- | --- |
| **내용** | [p.239] typecast.c | | |
| **코드** | #include <stdio.h>  int main()  {  int a = 3.4;  double d = 3;  printf("%5d %10f ", a, d);  printf("%10f\n", 3 + 4.5);  printf("%5d ", 10 / 4);  printf("%10f ", (double)10 / 4);  printf("%10f ", 10 / (double)4);  printf("%10f\n", (double)(10 / 4));    printf("%5d ", (int)(3.4 + 7.8));  printf("%10d ", (int) 3.4 + (int) 7.8);  printf("%10f ", (int) 3.4 + 7.8);  printf("%10f\n", 3.4 + (int) 7.8);  return 0;  } | | |
| **실행결과** |  | | |
| **고찰** | 계산을 함에 있어서 매우 중요한 형변환에 대하여 복습하였다.  형 변환 순서에 따라 값이 다르게 출력되는 것을 보면서 코드를 작성할 때 형변환 사용 여부, 순서를 매우 신중하게 결정해야 한다는 것을 느꼈다. | | |

## 6. [Ch. 5] 연산자 (2)

|  |  |  |  |
| --- | --- | --- | --- |
| **내용** | [p.245] priority.c | | |
| **코드** | #include <stdio.h>  int main(){  int a = 4, b = 6;  double x = 3.3, y = 4.7;  printf("%d ", a + b > y && x < y);  printf("%d ", a++ - --b \* 2);  printf("%f ", a > b ? x + 1 : y \* 2);  printf("%f ", x += 3 && y + 2);  printf("%f\n", (x = x + 1, y = y + 1));  return 0;  } | | |
| **실행결과** |  | | |
| **고찰** | 꼼꼼한 비교를 통하여   * 산술 > 관계 > 논리 * 단항 > 곱셈 > 뺄셈 * 산술 > 관계 > 조건 * 산술 > 논리 > 대입 * 괄호 > 산술 > 대입 > 컴마   의 순서로 실행 됨을 확인하였다. | | |

## 7. [Ch. 6] 조건

|  |  |  |  |
| --- | --- | --- | --- |
| **내용** | 세율 계산 | | |
| **코드** | #define \_CRT\_SECURE\_NO\_WARNINGS  #include <stdio.h>  int main(void)  {  int income,  overInc,  fee = 0;  float tax1  , tax2  , taxRate;  scanf("%d", &income);  if (income > 30000) {  fee = 9010;  taxRate = .38f;  overInc = income - 30000;  }  else if (income > 8800) {  fee = 1590;  taxRate = .35f;  overInc = income - 8800;  }  else if (income > 4600) {  fee = 582;  taxRate = .24f;  overInc = income - 4600;  }  else if (income > 1200) {  fee = 72;  taxRate = .15f;  overInc = income - 1200;  }  else {  fee = 0;  taxRate = .06f;  overInc = income;  }  tax1 = fee + overInc \* taxRate;  tax2 = income \* taxRate;  printf("과세o: %.0f만원\n", tax1);  printf("과세x: %.0f만원\n", tax2);  return 0;  } | | |
| **실행결과** |  | | |
| **고찰** | 소득 비용에 따른 서로 다른 세율을 계산하는 예제를 통해 if, else 조건문에 익숙해졌다. | | |

## 8. [Ch. 7] 반복

|  |  |  |  |
| --- | --- | --- | --- |
| **내용** | start 부터 end 까지의 합 | | |
| **코드** | #include <stdio.h>  int main(void)  {  int start, end, sum = 0;  printf("2개의 정수 (start, end)를 입력: ");  scanf("%d %d", &start, &end);  for (int i = end; i >= start; i--)  sum += i;  printf("%d부터 %d까지의 합: %d\n", end, start, sum);  return 0;  } | | |
| **실행결과** |  | | |
| **고찰** | for문을 활용하여 입력받은 start부터 end까지의 합을 구하는 코드를 작성하였다.  반복 연산자 중에서는 역시 for문이 가장 깔끔하고 이쁘다고 생각하였다. | | |

## 9. [Ch. 7] 반복 (2)

|  |  |  |  |
| --- | --- | --- | --- |
| **내용** | 나를 속이는 거짓말 | | |
| **코드** | #include <stdio.h>  int main()  {  char isLier;  int Innesim = 9;  printf("당신은 거짓말을 한 적이 있습니까? (Y/N)\n");  scanf("%c", &isLier);  while(1)  {  if (isLier == 'Y' || isLier == 'y')  {  printf("알고 있었어요.\n");  break;  }  if (Innesim < 0)  {  printf("거짓말 치지마요.\n");  break;  }  printf("진짜로 당신은 거짓말을 한 적이 없나요? (Y/N): ");  scanf(" %c", &isLier);  // 부정형으로 질문하였으니 한국 문법에 맞게 대답을 역전 시켜준다.  isLier = (isLier == 'Y' || isLier == 'y') ? 'N' : 'Y';  Innesim--;  }  return 0;  } | | |
| **실행결과** |  | | |
| **고찰** | 반복문을 통해 원하는 입력을 받을 때 가지 질문을 반복하는 재밌는 프로그램을 만들었다.  삼항 연산자 및 조건문, 반복문에 익숙해 졌음을 느꼈다. | | |

## 10. [Ch. 8] 배열

|  |  |  |  |
| --- | --- | --- | --- |
| **내용** | 빈도 출력 | | |
| **코드** | #include <stdio.h>  int main(void) {  int input;  int data[10] = { 0 };  int max = 0;  for (int i = 0; i < 20; i++)  {  scanf("%d", &input);  data[input]++;  if (data[max] < data[input])  max = input;  }  printf("가장 많이 입력받은 수는 %d이고 그 빈도는 %d이다.\n", max, data[max]);  return 0;  } | | |
| **실행결과** |  | | |
| **고찰** | 배열을 선언하고 값을 0으로 초기화하여, 증감연산자를 통해 빈도수를 계산하는 프로그램이었다.  입력받을 수 있는 최대 숫자에 해당하는 크기만큼 선언되어야 하고, 사용되지 않는 원소가 있을 수 있다는 단점을 해결 할 수 있으면 좋겠다고 생각하였다. | | |