

새싹교실

새싹교실

Week3

- 숙제 풀이 (간단한 알고리즘)
- C Language (조건문, 반복문)

고주형

2019/4/11

Homework Check

Homework Check

Week2 숙제 1

- 1부터 100까지 홀수만 더하고 출력하는 프로그램 만들어 보자!

(자기가 계산하면 안됨!!)

```
void main() {  
    int i = 0;  
    int sum = 0;  
  
    while (i<=100)  
    {  
        if(i%2==1)  
            sum += i;  
        i++;  
    }  
    printf("%d\\n", sum);  
}
```

```
void main() {  
    int i = 1;  
    int sum = 0;  
  
    while (i<=100)  
    {  
        sum += i;  
        i = i + 2;  
    }  
    printf("%d\\n", sum);  
}
```

Week2 숙제 2 고민해보기

- 1부터 100까지 중에서 소수만을 출력해보자!



```
#include <stdio.h>
```

```
void main() {  
    //소수를 판별하기 위한 변수 i 선언  
    int i = 2;  
    // i를 100까지 반복  
    while (i<=100)  
    {  
        // i가 인자를 가지고 있는지 판별하기 위한 j를 선언  
        int j = 2;  
        // 인자를 가지고 있는지 체크하기 위한 변수 hasDiv선언  
        int hasDiv=0;  
  
        // i가 소수인지 판별하기 위해 2부터 i-1까지 j를 반복시킨다.  
        while (j < i) {  
            //만약에 i보다 작은 수로 나뉘지면 인자가 있다고 hasDiv에 참을 넣음. (0이 아니면 참임을 이용.)  
            if (i%j == 0) {  
                hasDiv++;  
            }  
            j++;  
        }  
  
        //인자를 가지고 있지 않으면 출력  
        if (!hasDiv) {  
            printf("%d\n", i);  
        }  
        i++;  
    }  
}
```

Week2 숙제 2 심화!!!!!!

굳이 내가 판별하고자 하는 숫자 i 보다 작은 숫자로 다 나눠봐야 될까?

- (레벨1) 2부터 나누니까 i 의 반 까지만 나누면 되겠네!
어차피 뒤는 중복되는 연산이니까,
16을 소수인지 판별해보자: $8*2$ 체크랑 $2*8$ 체크랑 같음!
- (레벨2) i 의 반까지? 루트 i 까지 나눠봐도 될 것 같은데? $16 = 2*8, 8*2, 4*4$
제곱근(나눌 수 있는 최소 사이즈의 수) 까지만 체크하면 되네!
- (레벨3) 뭐야뭐야 굳이 하나하나 다 판별해? 난 몽땅 다 한꺼번에 처리 할래!
- 에라토스테네스의 체

Week2 숙제 3

- 1부터 100 더하기(2가지 방법)
- A--, A++ (즉, 탈출 조건을 다르게)



A++

```
#include <stdio.h>

void main() {
    int i, sum;
    i = 0;
    sum = 0;

    while (i < 101) {
        sum += i;
        i++;
    }

    printf("Sum = %d", sum);
}
```

A--

```
#include <stdio.h>

void main() {
    int i, sum;
    i = 100;
    sum = 0;

    while (i > 0) {
        sum += i;
        i--;
    }

    printf("Sum = %d", sum);
}
```


오늘 할 것

- 조건문 (if, switch)
- 논리 연산자(&&, ||, !)
- 반복문 (while, do~while, for)
- 복합대입 연산자 (+=, -=, *=, /=)
- 증감연산자(후위, 전위)
- 점프문 (break, continue, goto)
- 중첩 반복문

조건문 - if

```
if (조건1){  
    //조건1 이면 실행  
}else if(조건2){  
    //조건1 아니고 조건2이면 실행  
}else{  
    //조건1,2 모두 아니면 실행  
}
```

조건문 - switch~case

```
if (조건1){  
    //조건1 이면 실행  
} else if(조건2){  
    //조건1 아니고 조건2이면 실행  
} else if(조건3){  
    //...  
} else if(조건4){  
    //...  
} else if(조건5){  
    //...  
}  
.  
.  
.
```

너무 한 눈에 안들어와! ㅠㅠ

조건문 - switch~case

switch (변수)

```
{  
    case 숫자1:           //변수가 해당 숫자일 경우 진입!!  
        실행문;  
        break;          //밑에 케이스들도 실행되길 원하지 않으니 break!  
    case 숫자2:  
        실행문;  
        break;  
    default:  
        실행문;  
        break;  
}
```

★ switch문은 멈추지 않아요! 그래서 break해줘야 돼요!

★ 변수는 문자형이랑 정수형이랑 가능해요!

switch문 - 심화

★ switch문은 멈추지 않아요! 그래서 break해줘야 돼요!

- 이것을 활용할 수는 없을까?
- 1000원: 사탕 줌!
- 2000원: + 과자 줌!
- 5000원: + 음료 줌!
- 10000원: + 치킨 줌!

★ 변수는 문자형이랑 정수형이랑 가능해요!

- a를 입력하면 apple출력!
- b를 입력하면 banana출력!
- 그 외를 입력하면 아무것도 안출력!

논리 연산자

- && AND
- || OR
- ! NOT

!연산자 써 보기: 밑에꺼 해석 가능?
x의 배수 == if(!(number%x))

복합대입 연산자

그냥 단축키!

- $a += x$

- $a -= x$

- $a *= x$

- $a /= x$

$$a = a + x$$

$$a = a - x$$

$$a = a * x$$

$$a = a / x$$

증감연산자

단축키 + α

- `i++` : 후위 1 증가, 먼저 실행 후 증가
- `i--` : 후위 1 감소
- `++i` : 전위 1 증가, 먼저 증가 후 실행
- `--i` : 전위 1 감소

증감연산자 - 확인해 보자

```
printf(“후위 증가 %d\n”, i++);
```

```
printf(“전위 증가 %d\n”, ++i);
```

점프문

점프? 스킵하다, 뛰어들다

- break – 여기까지 그만 두고 나가!
- continue – 여기까지 그만 두고 계속해!
- goto – 여기로 가!

중첩 반복문

반복문 안의 반복문

- 구구단! (첫번째 수 따로 두번째 수 따로)
- 별 출력! (행 따로 열 따로)

*

**

복습 1

- a와 b를 Swap using variables

```
temp = a; //temp의 a의 값 임시로 저장  
a = b;  
b = temp;
```

복습 2

- 1~3까지 Sum 구하기

```
int i, sum = 0;
while(i<4)
{
    sum += i;
    i++;
}
```

숙제 1

- 다이아몬드(7줄 짜리 정도) 출력해보기.
(그냥 printf여러번 쓰면 안됨!!)
-

```
*  
***  
*****  
*****  
*****  
***  
*
```


숙제 2 고민해보기

- 1부터 100까지 중에서 소수만을 출력해보자!
- 효율적으로 다시 한번 만들어 보기



숙제 3

- do while문을 while문으로 바꿔보자.
- while문을 for문으로 바꿔보자



Quiz

C언어는 고급 언어? 저급 언어?

저급 언어는 어떤 언어?

