

뒤에 시험에 나올 것 같은 확률 적어드릴게요

1. #include -> 전처리 지시문 60%

2. <stdio.h> -> 헤더 파일 60%

#include <stdio.h> -> 이걸 종합하면 헤더파일 선언문

3. 산술연산자에서 100%

+, -, *, / 는 알고 있겠지만 %는 잊을 수 있으니 까먹지 말기

4. 함수의 구조 90%

반환타입 함수명(인자타입 인자명){

 함수 실행식

 return 반환할거

}

단, 여기서 반환값이 없는

반환타입을 void 함수면 return하지 않아도 됨.

return은 가능하지만 값 넣는 것은 안 됨.

함수

```
#include <stdio.h>
int sum(int a, int b);

void main() {
    int x, y, c;
    scanf("%d %d", &x, &y);

    .....
    c = sum(10, 20);
    printf("%d", c);
}

int sum(int a, int b) {
    int d;
    d=a+b;
    return(d);
}
```

→ 함수의 원형 선언

→ 함수의 호출

→ 함수의 정의

5. 진법변환

65%

정수 -> 2진수

2진수 -> 정수로

변환해보기

6. 데이터타입별 크기

70%

Int는 4바이트 등등

(진법변환 알면 int가 4바이트라는 것에서 몇부터 몇까지 숫자표현이 가능한지 확인 가능)

7. 오버플로, 언더플로 약간 알아만 보기 40%

오버플로 나도록 숫자 주고 값이 얼마인지 시험 나올 수 있음

➔ 음수를 2진법으로 표현하는 것도 연습해야 할지도

보통의 숫자형 타입은 (unsigned 사용하지 않았을 때)

맨 앞에 부호비트를 하나 가짐.

맨 앞의 부호가 0이면 양수, 1이면 음수

5 -> 0101

-5 -> 101에서 부호 반전해서 1010 -> 여기에 +1 하여 1011

8. 기계어와 어셈블리어는 저급언어 20%

c언어는 고급언어임(사람 기준 편한게 고급)

(자바, 파이썬 등 이후에 나온 언어는 대부분 고급언어)

9. 데이터 표기 20%

8비트 -> 1바이트

바이트, 킬로바이트, 메가바이트, 기가바이트, 테라바이트

이 순서로 대략 1000씩 곱하는 방식(정확히는 1024이나, 안 나올 듯)

10. 개발 단계

30%

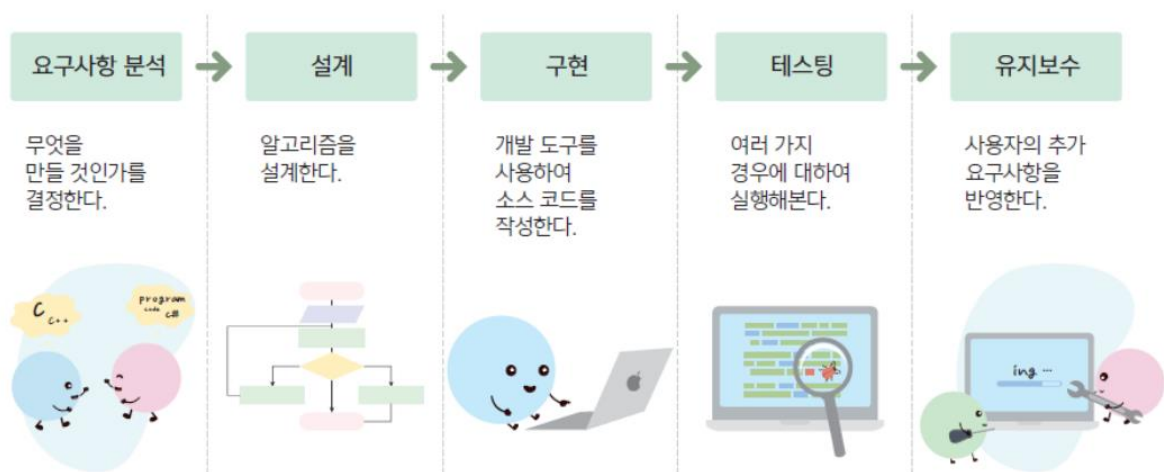
개발자 입장에서 개발 단계

프로그램 개발 단계



전체적인 프로젝트의 개발 단계

프로그램 개발 단계



11. 예약어

30%

변수와 같이 내가 직접 만들지 않아도 이미 언어에 있는거

예약어(reserved word)

자료형 관련 예약어

char, int, float, short, long, double, unsigned, union, enum, void, ...

기억 관련 예약어

auto, static, extern, register, ...

제어 관련 예약어

if~else, for, while, do~while, switch~case, break, continue, return, ...

기타 예약어

main, sizeof, include, ...

기억 예약어는 딱히 안 중요해 보임

제어 관련 예약어는 한마디로 설명하자면 프로그램의 순서를 바꿀 수 있는 예약어

(반복, 건너뛰기, 조건에 따른 작동 등)

기타예약어는 이미 c언어 자체에 선언되어 있는 함수들, 전처리 지시자 등

12. 변수 이름 규칙 60%

- 영어와 _(밑줄)과 숫자로 구성 (공백도 사용불가)
- 시작은 무조건 영어로 시작
- 예약어 사용불가(if, main, printf 등)

위 세 가지가 가장 중요하고 아래 사진은 참고만

명칭(identifier)을 만들 때의 규칙

- 영문자와 숫자의 조합으로 만든다.
- 명칭의 첫 문자는 영문자나 밑줄(_)이어야 한다.
- 특수문자를 사용해서는 안 된다.(단, 밑줄(_) 사용가능)
- 문자 사이에 공백이 있어서는 안 된다.
- 예약어를 사용할 수 없다.
- 영문자 대문자와 소문자는 서로 구별하여 사용
- 명칭의 길이는 컴파일러에 따라 차이가 있다.
(일반적으로 32자까지 인식가능)

13. 이스케이프 문자

50%

이스케이프 시퀀스(escape sequences)

- 화면상의 어떠한 상황 또는 상태를 표현하기 위해 약속된 문자

'\b'	백스페이스 문자
'\t'	탭 문자
'\\'	백슬래시 문자
'\''	작은따옴표 문자
'\"'	큰따옴표 문자
'\n'	개 행 문자
'\r'	캐리지 리턴(carriage return) 문자

코드 관련

- c언어는 ;가 작업처리의 기본단위
- // 는 한 줄 주석
- /* */ 로 감싸면 그 안에 것들이 주석처리

Ex) printf("결과는 %d", result); /* 이 코드는 결과를 출력하는 코드 */

- 함수, 조건문, 반복문의 범위는 중괄호 {}로 감싸진 코드블록에 적용
- 기호상수 define 따로 찾아보기

배열 진도 나갔으면 반복문이란 같이 사용해보기

별찍기로 반복문 많이 연습해보고

While과 for문 바꿔가며 작성해보기

2학기에 제대로 배울 것 같은거

1. 구조체
2. 포인터