

1. 함수

A. 함수의 정의

- i. 함수는 특정 작업을 수행하는 코드 블록

B. 함수의 기본 구조

C. 직무 관련 질문과 답변

- ii. 함수의 매개변수와 반환값?

매개변수 : 함수에 입력으로 전달되는 값

반환값 : 함수가 처리한 결과를 반환

2. 배열

A. 배열의 정의

- i. 동일한 데이터 타입의 연속적인 메모리 공간

B. 배열의 기본 문법

```
int speeds[5] = {60, 70, 80, 90, 100};
```

```
// 배열 요소 출력
```

```
for (int i = 0; i < 5; i++) {  
    cout << "Speed " << i + 1 << ": " << speeds[i] << " km/h" << endl;  
}
```

C. 직무 관련 질문과 답변

- i. 배열의 장점과 단점?

장점 : 여러 데이터 효율적으로 저장 및 접근

단점 : 고정된 크기 때문에 동적으로 크기 조정 X

3. 포인터

A. 포인터의 정의

- i. 변수의 메모리 주소를 저장하는 데이터 타입

- ii. 사용 예: HW와 SW 간 데이터 공유

B. 포인터의 기본 문법

```
int value = 42;

int* ptr = &value; // 포인터에 주소 저장

cout << "Value: " << value << endl;
cout << "Address: " << ptr << endl;
cout << "Dereferenced Value: " << *ptr << endl;

return 0;
```

C. 직무 관련 질문과 답변

i. 포인터란 무엇이며 왜 사용하나요?

포인터는 변수의 메모리 주소를 저장, 동적 메모리 할당이나 HW 데이터와의 상호작용에 사용.

예: 차량 ECU 간 데이터를 효율적으로 처리할 때 포인터 활용

ii. 포인터를 사용할 때 발생할 수 있는 문제?

잘못된 메모리 접근으로 인한 세그먼트 오류, 메모리 누수 등이 발생할 수 있음. 이를 방지하려면 포인터를 신중히 초기화하고 해제해야함.