



R E P O R T

제1장 프로그래밍의 개념

과목명	C 프로그래밍 I
분반	4 분반
교수	문 일 영
학번	2020136129
이름	최 수 연
제출일	2020년 4월 5일 일요일

01. 컴퓨터가 사용하는 진법은 무엇인가? (1)
02. 고급언어로 작성된 프로그램을 기계어로 바꾸어주는 도구는 무엇인가? (2)
03. 문제를 해결하는 절차를 시각적으로 표현한 것은 무엇인가? (2)
04. 다음 중 c언어의 특징으로 적합하지 않은 것은? (2)
05. 컴퓨터를 이용하여 문제를 해결하기 위한 절차를 무엇이라고 하는가? (1)
06. 알고리즘을 기술할 수 있는 방법을 모두 고르라. (1, 2, 3)
07. 순서도(flowchart)에서 처리를 나타내는 기호는? (2)
08. 다음 중에서 C언어를 개발한 사람은 누구인가? (1)

09. 컴퓨터에서 이미지와 음악이 어떻게 표현되는지를 인터넷에서 조사하여 보라.

먼저 이미지의 경우 실존하는 어떤 사물이나 장소를 이미지 센서를 통해 디지털 값들의 집합으로 표현한다. 이러한 데이터는 가장 간단한 형태인 이진수를 이용하여 표현이 되는데, 이진수들의 조합을 통해 이미지를 이루는 가장 작은 단위인 픽셀, 즉 화면에 표현되는 작은 점들이 모여 이미지를 만들어 낸다.

음악의 경우도 마찬가지로 컴퓨터는 이진법을 사용하기 때문에 음을 표현하는 과정에서 0과 1로 음계를 설정하여 원하는 음악을 만들어낸다.

10. 컴퓨터에서 10진법이 아닌 2진법이 사용되는 이유를 설명하라.

만약 컴퓨터가 2진법 이상을 기반으로 설계가 된다고 가정하면 데이터 처리 과정에서의 처리 시간이 길어질 뿐만 아니라 더 많은 비용이 발생하게 된다. 특히 현재는 2진법을 사용함에도 불구하고 데이터 노이즈 현상이 발생하는데, 2진법 이상을 이용하여 컴퓨터를 설계할 경우 지금보다 더 큰 노이즈가 발생할 것으로 예상된다. 그렇게 되면 데이터 송수신에 있어 비용이 증가하게 된다.

또한, 컴퓨터가 2진법으로 설계된 가장 명백한 이유는 컴퓨터 논리회로 자체가 전기 신호 유무에 따라 0과 1로 구분지어 설계되어있기 때문이고, 2진법을 이용해도 처리할 수 없는 값은 없기 때문에 컴퓨터에서는 2진법이 가장 적합하다고 판단한다.

11. 인텔의 CPU에서 사용되는 명령어 중에서 3가지를 선택하여 무슨 일을 하는 명령어인지를 조사하여 보라.

CISC(Complex Instruction Set Computing)

CISC는 사용자가 작성하는 고급언어에 각각 하나씩 기계어를 대응시킨 회로로 구성된, 중앙처리장치의 한 종류로써, 마이크로프로그래밍을 통해 고급 언어에 각기 하나씩의 기계어를 대응시킴으로써 명령어 집합이 커지고, 가변 길이의 다양한 명령어를 가진다. 즉, 쉽게 말해 명령어가 많고 명령어의 길이도 다양하기 때문에 속도가 느리다. 그러나 CISC는 하나의 프로세서가 일련의 명령어를 순차적으로 처리하기에는 무척 유용한 방법이고, CPU의 동작 속도가 높아짐에 따라 성능이 비례로 증가한다.

RISC(Reduced Instruction Set Computer)

RISC는 reduce 말 그대로 명령세트를 축소하여 설계되었기 때문에 CPU에서 수행하는 모든 동작의 대부분이 몇 개의 명령어만으로 가능하다. 이는 고정된 길이의 명령어를 사용하고 명령어의 종류가 미리 정해져 있어 해석 속도가 빠르고 여러 개의 명령어를 처리하기에 적합하다. 특히, 분기 위치가 정해져 있고 비순차 처리도 가능하다.

VLIW(Very Long Instruction Word)

VLIW는 여러 opcode 필드가 있는 긴 명령어 하나에 독립적인 연산 여러 개를 정의하고 이들을 한꺼번에 내보내는 명령어 구조 집합의 종류이다. 명령어가 일반적으로 128비트 또는 256비트 단위로 구성되어 있고, 경우에 따라 512비트까지도 사용된다. VLIW는 명령어가 처리 방식에 따라서 구분되는 것이 아니라 명령어 필드의 비트가 구분되므로 명령어 자체의 처리 단위가 좀 더 복잡해진다. 이 방식은 3차원 그래픽 처리나 멀티미디어 가속 등 비교적 단순한 자료의 처리가 반복되는 경우에 유용하다.

12. 주판, 계산자, 계산판 등의 고대의 계산하는 기계들과 현대의 컴퓨터와 결정적인 차이점은 무엇인가?

주판, 계산자, 계산판 등과 같은 고대의 계산하는 기계들의 특징은 아날로그 방식을 이용하여 계산하고, 현대의 컴퓨터 같은 경우에는 디지털 방식을 이용하여 계산을 한다.

13. 컴퓨터 부품을 판매하는 인터넷 쇼핑몰을 방문하여서 컴퓨터의 부품에는 어떤 것들이 있고 어떻게 분류할 수 있는지를 조사하라.

컴퓨터는 기본적으로 모니터, 키보드, 마우스, 본체 총 4가지로 구성되어 있다. 또한, 본체는 다양한 부품들로 구성이 되어있는데, 다음과 같이 구분할 수 있다.

1. CPU(중앙처리장치) - 트랜지스터의 집합체
2. RAM(임시저장장치) - CPU 우측, DDR3, DDR4
3. 메인보드 - 데이터버스, 소켓, 슬롯, 전력공급
4. 저장장치 (HDD, SSD) - 하드디스크(플래터, 암(Arm))
5. 파워서플라이(전원공급장치) - 전원공급 부품. 직류 전환 장치
6. 그래픽카드(VGA) - GPU, VRAM
7. CPU 쿨러
8. PC케이스 - 슬림, 미니타워, 미들타워, 빅타워

14. 컴퓨터를 이용하여 문제를 해결할 때, 문제를 해결하는 방법을 인간이 구체적으로 고안하여 알려주어야 한다면 왜 컴퓨터를 사용하는 것인가? 인간이 직접 하는 편이 낫지 않을까? 컴퓨터를 사용하는 경우의 장점을 들어 보라.

컴퓨터와 같은 기계는 인간이 할 수 있는 일을 대신하여 시간을 단축시킴으로써 인간이 좀 더 능률적으로 일을 할 수 있게 돕는다. 즉, 어떠한 문제를 해결하기 위해서 인간이 구체적으로 그 과정을 고안하는 것과 그 문제 해결 방법을 인간이 직접 풀어나가는 시간을, 컴퓨터가 인간이 구체적으로 고안한 알고리즘을 주어주면 이에 대한 계산을 대신하여 결과를 빠르게 처리해 알려줌으로써 인간은 시간을 더욱 효율적으로 사용할 수 있게 된다.

(문제를 해결하는 방법을 인간이 구체적으로 고안하여 알려주어야 한다는 점에서 컴퓨터를 이용하는 것의 한계가 있긴 하지만, 인간은 꾸준히 과학 기술을 발전시키고 있기 때문에, 언젠가는 인공지능을 통해 문제 해결 방법에 대한 고안까지 모두 맡길 수 있지 않을까 생각한다.)

15. 기계어, 어셈블리어, 고급 언어의 차이점을 정리하여 보라.

기계어, 어셈블리어, 고급 언어 순대로 봤을 때, 기계어 쪽으로 갈수록 기계에 더 가까우며, 고급언어로 갈수록 인간에게 더 가까운 언어라고 볼 수 있다. 또한, 기계어와 어셈블리어는 저급 언어에 속하는데, 기계어에서 숫자를 의미 있는 단어로 바꾸어 사람들이 이해하기 쉽게 만든 언어가 어셈블리어 이므로 기계어보다는 어셈블리어가 좀 더 인간에게 가까운 언어라고 볼 수 있다. 기계어는 컴퓨터가 사용하는 언어이고, 고급언어는 C언어처럼 인간들이 사용하는 언어이기 때문에 C언어와 같은 고급언어로 작성된 명령어를 컴퓨터가 이해하기 위해선 컴파일러를 이용해 기계어로 바꾸어주어야 한다.

16. C++와 Java는 C에다 어떤 개념을 추가한 것인가? 인터넷에서 검색하여 정리하여 보라.

C언어는 절차 지향적(순차 지향적) 언어를 기본으로 하는데, C++와 Java는 C언어에 객체 지향적인 특징들을 추가한 언어이다. 즉, C++과 Java는 C언어의 확장판이라고 할 수 있다.

17. 임베디드 시스템이란 어떤 것인가? 인터넷에서 자료를 찾아서 정리하여 보라.

임베디드 시스템이란 특수 목적의 시스템으로, 어떤 제품이나 솔루션에 추가로 탑재되어 그 제품 안에서 특정한 작업을 수행하도록 하는 솔루션을 말한다. 예를 들어 스마트폰의 음악을 듣는 MP3의 기능이나 스마트폰의 길 찾기 앱에서의 내비게이션 기능 등과 같이 본 시스템에 다른 시스템의 기능이 추가된 것 모두 임베디드 시스템이라 할 수 있다.

18. 아이폰과 안드로이드폰에서는 어떤 언어를 사용하여서 애플리케이션을 개발하는지 조사하여 보라.

아이폰 애플리케이션 개발할 때 사용되는 언어는 보통 Objective-c 이고, 안드로이드 폰의 애플리케이션 개발할 때 사용되는 언어는 JAVA나 Kotlin를 사용한다.

19. 다음과 같은 일상적인 행위에 대한 알고리즘을 작성하여 보라.

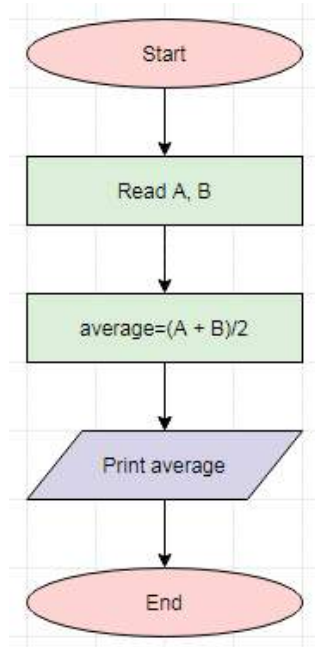
(a) 프린터를 이용하여 인쇄를 한다.

1. 프린터의 전원을 켜다.
2. 용지 유무를 확인하고 없으면 원하는 크기의 용지를 프린터에 삽입한다.
3. 토너 유무를 확인하고 없으면 토너를 교체한다.
4. 프린트 하고자 하는 데이터의 화면에 있는 인쇄 아이콘을 누른다.
5. 인쇄 범위, 방식, 매수 등을 원하는 대로 선택한다.
6. 인쇄 버튼을 누른다.
7. 출력된다.

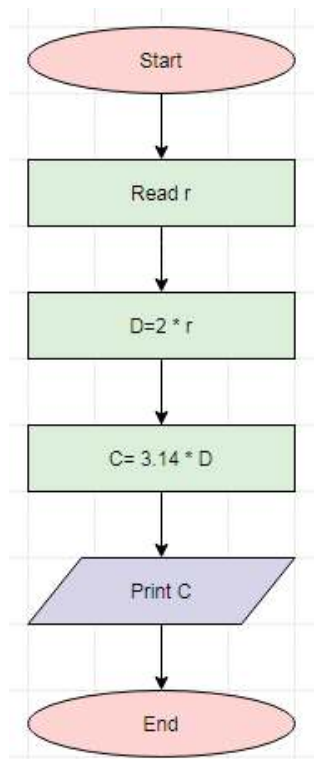
(b) 인터넷 쇼핑몰에서 상품을 구입한다.

1. 구매하고자 하는 상품을 파는 쇼핑몰을 선택한다.
2. 선택한 인터넷 쇼핑몰에 들어가 검색창에 구매하고자 하는 상품의 이름을 작성한다.
3. 검색버튼을 누른다.
4. 상품을 클릭한다.
5. 구매하기 버튼을 누른다.
6. 상세 주소 및 결제 수단 등을 입력한다.
7. 선택한 결제 수단에 맞게 결제 절차를 밟는다.
8. 결제가 완료되면 상품 주문이 되었는지 확인한다.
9. 원하는 배송지에 배송이 완료된다.

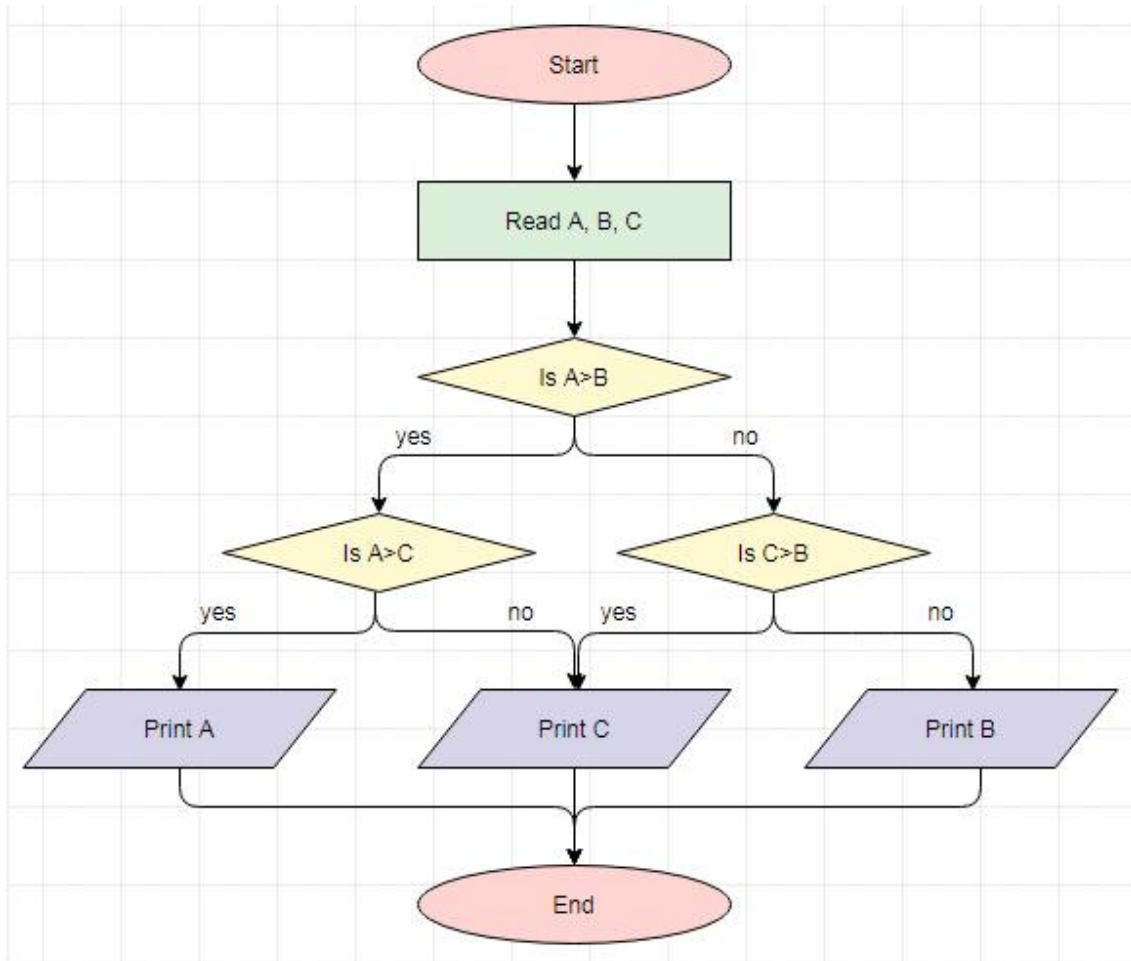
20. 주어진 두 수의 평균을 구하는 알고리즘을 순서도로 작성하여 보라.



21. 사용자로부터 원의 반지름을 입력받고 반지름에 2를 곱하여 지름을 구하고 여기에 3.14를 곱하여 원의 둘레를 계산하는 알고리즘을 순서도를 이용하여 기술하라.



22. 두 개의 숫자 중에서 큰 수를 반환하는 연산만 지원되는 컴퓨터가 있다. 이 컴퓨터에서 3개의 숫자 중에서 제일 큰 수를 찾으려고 하면 어떤 알고리즘을 사용해야 하는가? 순서도로 그려보자.



23. 두 개의 컵에 우유와 주스가 각각 담겨있다. 우유와 주스를 교환하기 위한 알고리즘을 고안하라. 사용가능한 세 번째 컵이 있다고 가정한다.

[첫 번째 방법]

1. 우유가 담긴 컵을 A, 주스가 담긴 컵을 B, 빈 컵을 C라고 가정한다.
2. 컵A에 담겨있는 우유를 컵C에 옮긴다.
3. 컵B에 담겨있는 주스를 컵A에 옮긴다.
4. 컵C에 담겨있는 우유를 컵B에 옮긴다.

[두 번째 방법]

1. 우유가 담긴 컵을 A, 주스가 담긴 컵을 B, 빈 컵을 C라고 가정한다.
2. 컵B에 담겨있는 주스를 컵C에 옮긴다.
3. 컵A에 담겨있는 우유를 컵B에 옮긴다.
4. 컵C에 담겨있는 주스를 컵A에 옮긴다.

24. 다음 알고리즘을 구상하여 보라.

(a) 만약 숫자들의 리스트가 주어지고 이 중에서 특정한 숫자를 찾는 알고리즘을 구상하여 보시오. 숫자들은 정렬되어 있지 않다고 가정하라.

숫자들의 리스트가 무작위로 배열되어 있다면, 숫자 하나씩 차례대로 찾고자하는 특정 숫자와의 비교를 통해 값을 찾아낸다.

차례대로 선택하여 그 하나의 숫자가 찾고자 하는 숫자와 비교하였을 때, 작거나 크면 다음 숫자로 넘어간다.

만약 무작위의 배열된 숫자들의 리스트 중에 순차적으로 비교를 수행할 때, 하나의 숫자가 찾고자 하는 값과 일치하면 위 알고리즘은 종료된다.

(b) 만약 숫자들이 크기순으로 정렬되어 있다면 특정한 숫자를 찾는 알고리즘을 어떻게 개선시킬 수 있는가?

만약 숫자들이 오름차순으로 정렬되어 있으면, 먼저 전체 데이터에서의 중앙값을 선택하고, 중앙값과 특정한 숫자, 즉 찾고자 하는 값과 비교한다.

만약 중앙값이 찾고자 하는 값보다 크면 중앙값의 왼쪽 부분, 즉 중앙값보다 작은 숫자들의 집합으로 범위를 축소한다.

만약 중앙값이 찾고자 하는 값보다 작으면 오른쪽 부분, 즉 중앙값보다 큰 숫자들의 집합으로 범위를 축소한다.

위 단계를 끝내면 다시 축소된 범위의 숫자들의 집합으로 중앙값과 찾고자 하는 값을 비교하여 반복 수행한다.

이후 중앙값과 특정한 숫자의 값이 같아지면, 이 알고리즘을 끝낸다.

25. 1부터 100사이의 정수 중에서 소수(prime number)를 찾는 알고리즘을 생각할 수 있는가?

소수가 되려면 1과 자기 자신만을 약수로 해야 하므로, 약수가 2개인 숫자들을 찾으면 된다. 자기 자신의 숫자를 i 라고 한다면, 1부터 i 까지의 정수를 나누어 나머지가 0이 되는 숫자의 개수가 2개인 경우의 i 만을 출력한다.

먼저, 정수 i 와 $n=0$ 을 선언한 뒤, 약수의 개수를 n 이라고 가정하면,

$i=1$ 부터 $i \leq 100$ 까지로 범위를 두고 $i++$ 하여 하나씩 증가하며 반복 수행하도록 설정한다.

약수의 개수는 2개일 때 i 를 출력하도록 설정한다.

(※여기서 약수의 개수가 1개이거나 3개 이상으로 설정하면 안 된다.)

1의 경우, 1부터 1까지의 정수를 나누어 나머지를 구하면,

$1\%1=0$ 이고, 나머지가 0일 때 약수의 개수 n 에 1을 더한다.

약수의 개수가 1개이므로 i 인 1을 출력하지 않는다.

2의 경우, 1부터 2까지의 정수를 나누어 나머지를 구하면,

$2\%1==0$ 이고, 나머지가 0일 때 약수의 개수 n 에 1을 더한다.

$2\%2==0$ 이고, 나머지가 0일 때 약수의 개수 n 에 1을 더한다.

약수의 개수가 2개이므로 i 인 2를 출력한다.

3의 경우, 1부터 3까지의 정수를 나누어 나머지를 구하면,

$3\%1==0$, $3\%2==1$, $3\%3==0$ 이고, 약수의 개수가 2개이므로 i 인 3을 출력한다.

위 과정을 100까지 반복 수행하여 알고리즘을 종료하고, 최종적으로 출력된 소수라고 판정된 i 의 값들을 확인한다.