**컴퓨터알고리즘과실습**

**[1주차 실습]**

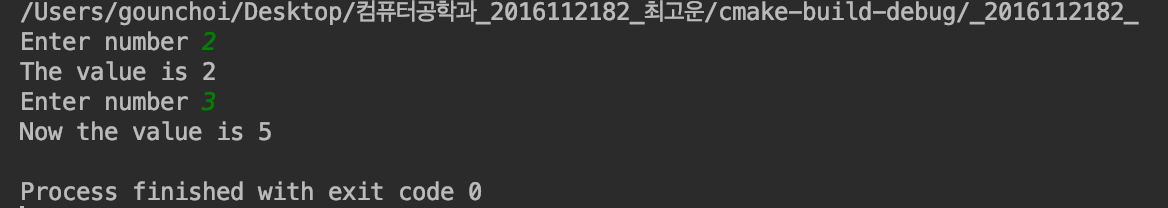
**컴퓨터공학과**

**2016112182**

**최 고 운**

* **1번문제**

**다양한 입력 값들을 사용하여 다음 프로그램을 실행해보고, 프로그램의 동작 원리를 설명하시오. 또한, 입력으로 2, 3을 사용하였을 때의 결과를 출력하시오.**



|  |
| --- |
| **소스코드 설명** |
| - 프로그램이 실행되고 main()함수가 실행되면 num이라는 이름의 InputNum 객체가 생성되면서 생성자가 실행된다. 생성자 안에서 num 객체의 멤버변수인 \_num에 정수를 하나 입력받아 저장한다.  - 생성자 실행이 완료되면 num객체의 GetValue() 함수가 실행되면서 num객체의 \_num변수에 저장된 값을 받아와 출력한다.  - num객체의 AddInput() 함수가 실행되면서 aNum이라는 이름의 새로운 InputNum 객체가 생성된다. aNum의 생성자가 실행되면서 위와 마찬가지로 생성자 안에서 값을 입력받아 aNum 객체의 멤버 변수 \_num에 값을 저장한다. 후에 num객체의 \_num에 aNum객체의 \_num값을 GetValue() 함수를 통해 받아와 더하여 다시 저장한다.  - 마지막으로 새롭게 저장된 num 객체의 \_num 값을 GetValue() 함수를 통해 받아와 출력한다. |
| **실행 예시 설명** |
| - num 객체의 멤버변수 \_num에 정수 2가 저장되고, num.GetValue()가 실행되면서 2라는 저장된 값을 받아와 출력한다.  - num 객체의 AddInput() 함수가 실행되면서 새로운 객체 aNum이 생성되고, 이의 생성자가 실행되면서 aNum객체의 멤버변서 \_num에 정수 3이 저장된다.  - AddInput() 함수 안에서 num 객체의 \_num 변수에 저장된 2에 aNum 객체에 저장된 3을 GetValue()함수를 통해 받아와 두 값을 더하여 다시 num 객체의 \_num변수에 5라는 값이 저장된다.  - num.GetValue()가 실행되며 5라는 값을 받아와 출력한다. |

* **2번문제**

**다음은 배열을 이용하는 스택(Stack1)의 응용 예이다. 다양한 입력 데이터에 대해 아래 프로그램을 실행해보고, 어떤 기능을 하는 프로그램인지 설명하시오. 또한, 아래의 입력 데이터를 사용하였을 때의 결과를 출력하시오.**



|  |
| --- |
| **소스코드 설명** |
| - 다양한 자료형의 stack을 쉽게 사용하기 위해서 typedef를 이용하여 itemType을 정의해준다. 주어진 실행 예시는 int형 stack이므로 int형으로 지정해 주었다.  - 클래스 분석   1. 멤버 변수 : itemType형 배열 stack과 int형 변수 p를 private로 선언해 주었다. 2. 생성자 : Stack1객체가 생성되면 stack배열을 max크기의 배열로 할당해준다. 이 때 max의 기본 값은 100이다. 또한 변수 p의 값도 0으로 초기화 해준다. 3. 소멸자 : Stack1객체가 소멸될 때 stack에 할당된 메모리를 해제시킨다. 4. 멤버 함수 : 각 함수가 간단하며, 짧기에 인라인 함수로 선언하여 인라인 함수의 이점을 사용하였다. 우선 push함수는 stack배열의 인덱스 p 위치에 매개변수로 받아온 itemType의 변수 v를 저장한 후 p의 값을 증가시키는 함수이다.   다음으로 pop함수는 p의 값을 감소시킨 후 stack배열에 인덱스 p 위치에 저장된 값을 반환하는 함수이다.  empty함수는 stack이 비어있으면 1(true)를 그렇지 않으면 0(false)를 반환하는 함수이다  - char형 변수 c를 선언해주고, Stack1 객체 acc를 생성한다. 이 때 생성자가 실행되면서 acc의itemType형 stack배열의 사이즈가 50으로 할당된다. 그리고 itemType형 변수 x를 선언한다.  - while문 분석   1. while문은 c에 개행 문자가 입력될 때까지 반복된다. 2. x는 Stack2객제 acc에 push할 값을 저장하는 변수로 처음에는 0으로 초기화한다. 3. c에 공백이 입력되었을 경우 계속 새로운 값을 입력 받아 c에 저장한다. 4. c가 +혹은 \*의 연산 기호일 경우 acc에 가장 끝에 저장된 두 값을 pop해 연산한 값을 x에 저장한다. 5. c가 한 자리 숫자일 경우 while문을 한 번 돌며 x에 int형으로 변환되어 저장된다. c가 두 자리 숫자일 경우 while문을 두 번 돌며 십의 자리 먼저 x에 아스키코드 값을 이용하여 정수형으로 변환되어 저장되고, 그 다음 일의 자리까지 변환되어 더해져 최종적으로 x에 입력 받은 두 자리 수가 정수형으로 저장된다. 6. 그리고 최종적으로 x에 저장된 값이 push된다.   - 개행 문자가 입력되어 위의 while문을 빠져나오면 acc객체의 pop이 실행되어 반환된 값을 출력하며 프로그램이 종료된다. |
| **실행 예시 설명** |
| - 문자형 변수 c와 Stack1의 객체 acc, 정수형 변수 x가 선언된다. 이 때 acc가 생성되면서 생성자가 실행된다. Int형 stack배열이 사이즈 50으로 할당되고 p는 0으로 초기화된다.  - while문에 들어가며 c에 첫 문자인 9를 입력 받고 그 안의 while문으로 들어가 x에 9가 저장된다 후에 acc.push(9)가 실행되며, stack[0]에 9가 저장되고 p의 값은 1이 된다.  - 다음으로 두 자리 수인 10을 입력 받고, 그 안의 while문으로 들어가면 c에 저장된 1이 정수형으로 변환되어 x에 저장되고, 다시 while문에 들어가면 x에 저장된 1에는 10이 곱해지고 c에 저장된 0은 정수형으로 변환되어 더해져서 결과적으로 x에 정수 10이 저장된다. 그리고 push함수가 실행되며, stack[1]에 10이 저장된다 .  - 위와 같은 과정으로 stack[2]에 15, stack[3]에 4가 저장되고 p의 값은 4가 된다.  - 후에 \*연산자가 입력되며 if문이 실행되고, pop함수가 실행된다. 그러면 p의 값은 3이 되고, stack[3]에 저장된 4가 반환되고, 이와 같은 과정을 통에 15가 반환 된 후 두 값을 곱한 60이 x에 저장된다. 이 때 p의 값은 2이다. 그리고 x에 저장된 60이 stack[2]에 저장되고 p의 값은 3이 된다.  - 위와 같은 과정을 통해 연산자를 만날 때마다 가장 나중에 저장된 값들이 반환되어 연산되는 과정이 반복되며 다음과 같은 과정을 통해 결과값이 출력된다.  - 9 10 15 4 \*  - 9 10 60 +  - 9 70 20 \*  - 9 1400 7 +  - 9 1407 +  - 1416 |

* **3번문제**

**아래 코드를 기반으로 포인터를 이용하는 스택(Stack2) 자료구조를 완성하고, 이를 통해 후위(postfix) 표기의 수식을 계산해주는 프로그램을 작성하시오. 단, +와 \*뿐만 아니라 -와 /까지도 연산이 가능해야 한다.**

**또한, 다음 입력 데이터를 사용하였을 때의 결과를 아래의 입출력 예제에 맞춰 출력하시오.**

개체, 시계, 게이지이(가) 표시된 사진

자동 생성된 설명

|  |
| --- |
| **소스코드 설명** |
| - 다양한 자료형의 stack을 쉽게 사용하기 위해서 typedef를 이용하여 itemType을 정의해준다. 주어진 실행 예시는 int형 stack이므로 int형으로 지정해 주었다.  - 클래스 분석   1. 멤버 변수 : node라는 이름의 구조체가 정의되어 있다. 구조체의 멤버는 itemType형의 변수 key와 node구조체 타입의 next포인터이다. 이 때 next 포인터는 다음 node의 주소값을 저장한다. 그리고 node구조체 타입의 head, z 포인터가 선언되어 있다. 본인은 head node를 현재 node를 가리키는 index 역할을 하는 node로 사용하였고, z node를 가장 처음 위치를 잡아주는 node로 사용하였다. 2. 생성자 : Stack2 객체가 생성되면 z node와 head node에 메모리를 할당하여주고, head의 next에 z의 주소값을 저장한다. 3. 소멸자 : head와 z node의 메모리를 해제한다. 4. 멤버 함수 : 먼저 push함수는 itemType의 v를 매개변수로 가진다. 함수가 실행되면 새로운 node구조체 타입의 ptr포인터를 선언하고 메모리를 할당해준다. ptr node는 현재 들어온 값을 담는 역할로 사용하였다. 그러므로 ptr의 key값을 v로 지정해주고 ptr의 next 포인터가 가리키는 값은 기존에 head node의 next가 가리키고 있던 기존 가장 최근의 node를 가리킨다. 그리고 head node의 next는 방금 추가한 ptr node가 가장 최근 node임으로 ptr node의 주소값을 저장해준다.   pop함수는 가장 최근의 값을 반환해주는 함수로 itemType의 변수에 head node의 next가 가리키고 있는 가장 최근 node의 key값을 저장해 반환한다. 또한 반환하면 가장 최근 노드가 그 전 노드로 업데이트 되어야 하기 때문에 가장 최근 node 즉 head node의 next가 가리키고 있는 노드의 next가 가리키고 있는 노드를 head node의 next에 저장해준다.  empty함수는 함수명 그대로 비어있는 스택인지 판단해주는 함수로, 텅 빈 스택이라면 head node의 next가 z node를 가리키고 있을 것이다. 이럴 경우 1(true)을 반환하고, 그렇지 않을 경우 0(false)을 반환한다.  - 모두 위의 2번 문제와 같은 원리로 실행되며, 빼기와 나누기 연산이 추가되었는데 이 연산들은 교환법칙이 성립되지 않는 연산으로 순서가 중요하다. 명시되어 있는 조건이 따로 없어 연산 결과로 유추한 결과 두 값을 pop했을 때 더 나중에 pop된 값, 즉 먼저 입력한 값이 앞에 위치한다고 판단하여 먼저 pop된 값을 num2변수에 저장, 나중에 pop된 값은 num1변수에 저장하여 num1변수에서 num2변수를 빼거나 나누는 방식으로 추가해 주었다. |
| **실행 예시 설명** |
| - 실행 예시 또한 2번의 문제와 흡사하다. 스택이 array가 아닌 linked-list로 구현되었으니 그 부분 위주로 작성한다.  - 20을 입력 받으면 2번에 설명한 방법과 같은 방법으로 정수형 20으로 변환되어 x변수에 저장되고 이 값이 push함수의 매개변수로 들어간다. 그러면 새로운 node ptr이 생기고 ptr의 key값이 20이 된다. Ptr은 head가 가리키고 있던 첫 노드 z를 가리키고 head는 새로 들어온 ptr노드를 가리키게 된다. 다시 15를 입력 받으면 새로운 공간을 할당하여 ptr이 생성되고 이 노드의 Key값이 15가 되고 이 노드는 head 노드가 가리키고 있던 20이 저장된 노드를 가리키고 head는 15가 저장된 ptr노드를 가리키게 된다. \*연산자가 입력되면 pop함수가 실행되면서 head가 가리키고 있는 노드의 key값인 15가 반환되고 head는 다시 20이 있는 노드를 가리키게 된다. 같은 과정으로 20도 반환되어 두 값이 연산된 결과가 다시 x에 저장되어 push된다  - 위와 같은 과정을 통해 연산자를 만날 때마다 가장 나중에 저장된 값들이 반환되어 연산되는 과정이 반복되며 다음과 같은 과정을 통해 결과값이 출력된다.  - 20 15 \*  - 300 20 4 /  ( 이 과정에서 먼저 pop된 4가 num2에 저장되고 그 다음 pop된 20이 num1에 저장되어 20/4 = 5 )  - 300 5 +  - 305 8 \*  - 2440 5 –  - 2435 |

* **4번문제**

**두 개의 자연수 N, M을 입력 받은 뒤 random 함수를 통해 N개의 자연수 집합 A와 M개의 자연수 집합 B를 만들어내는 프로그램을 작성하시오. 또한, 두 집합 A와 B의 합집합과 교집합을 아래의 예제와 같이 출력해주는 프로그램을 작성하고 그 결과를 출력하시오. 마지막으로 합집합과 교집합을 찾아내는데 사용된 알고리즘의 원리를 설명하시오.**

**(단, M,N ≥10 이고 각각의 집합 안에서는 중복되는 값이 존재하지 않는다고 가정한다)**

텍스트이(가) 표시된 사진

자동 생성된 설명

|  |
| --- |
| **소스코드 설명** |
| - random수의 범위를 정하는 데 사용할 값을 변경할 경우 편의성을 위해 문자 상수 MAX로 정의해 주었다.  - 집합의 크기가 정해지지도 않았고, 합집합을 구현할 때 한 집합에서 교집합을 삭제해주는 아이디어를 사용할 예정이기에 vector를 사용하는 것이 가장 효율적이라고 판단하였다.  - intersectionset함수 : a 벡터를 기준으로 for문을 돌며 a벡터의 각 원소와 b벡터으 ㅣ  - unionset함수 :  - displaySet함수 :  - |
| **실행 예시 설명** |
|  |

* **소감**