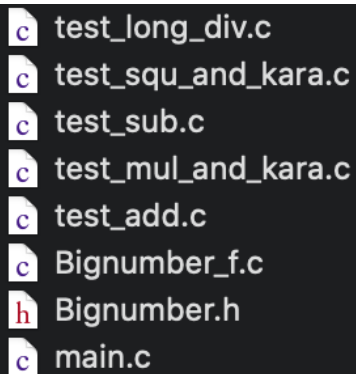

Advanced Application Programming

프로젝트 사용법

20175204 김 영 범



```
c test_long_div.c
c test_squ_and_kara.c
c test_sub.c
c test_mul_and_kara.c
c test_add.c
c Bignumber_f.c
h Bignumber.h
c main.c
```

제 프로젝트는 총 8개의 파일로 이루어져 있습니다. Bignumber.h 파일은 프로젝트 함수들을 모아둔 헤더파일이며, Bignumber_f.c 는 수업시간에 배운 함수들을 구현하여 모아둔 파일입니다. (총 2개)

이름 앞에 test가 붙어있는 파일들은 중요 연산 덧셈, 뺄셈, 곱셈(카라추바 포함), 제곱(카라추바 포함), 그리고 나눗셈(Bit_version)에 대하여 Verification 및 performance를 확인 하는 함수로 이루어져 있습니다. (총 5개)

마지막으로는 main.c 파일입니다. 이 파일은 두수 혹은 세수를 만들어, 지금까지 제가 구현한 모든 함수를 하나씩 실행시킬 수있게 만든 파일입니다.

test_add.c 파일중 Verification에 해당하는 부분입니다. 하얀색 박스안에 있는 숫자들은 생성하고 자 하는 Wordlen 이며, 노란색 박스안에 있는 숫자는 검증할 횟수 입니다. 변수는 하얀박스, 노란박스 두가지 입니다.

```
int main() /// Addition이 잘 작동되는지 확인하는 함수 (Verification)
{
    int i = 1000 // 몇번 확인할지 값 대입
    srand(time(NULL));
    while ((i >= 0))
    {
        bigint_st_ptr A = NULL; // 변수 생성
        bigint_st_ptr B = NULL;
        bigint_st_ptr C = NULL;
        i--;
        generate_random_integer(&A, NON_NEGATIVE, 100) // 임의의 숫자 생성
        generate_random_integer(&B, NON_NEGATIVE, 101) // 임의의 숫자 생성
        printf("a = ");
        show(A);
        printf("b = ");
        show(B);

        Addition(&A,&B,&C);
        printf("c = ");
        show(C);
        printf("\n");
        // 파이썬 문법으로 작성하였다. 컴파일 하면 파이썬에서 내가 확인하고 싶은 데이터들의 개수만큼
        // 파이썬문법으로 조건문이 나온다

        // printf("if a + b == c :\n"); /// * 생략가능
        // printf("    print \"True\"\n");

        printf("if a + b != c :\n");
        printf("    print hex(a)\n");
        printf("    print hex(b)\n");
        printf("    print hex(c)\n");
        printf("    print hex(a+b)\n");

        delete_bigint(&A); // 메모리 동적할당 해제
        delete_bigint(&B);
        delete_bigint(&C);
    }
    return 0;
}
```

test_add.c 파일중 Performance에 해당하는 부분입니다. 하얀색 박스안에 있는 숫자 int i는 addition 기준으로 몇번의 addition을 수행할건지 설정해주는 변수이고, size1은 addition함수의 input data A값을 생성할 랜덤 wordlen을 size2는 B값을 생성할 랜덤 wordlen을 의미합니다. 변수는 총 세 개 입니다.

```
int main()
{
    clock_t start, end;
    double cpu_time_used1;
    double cpu_time_used2;

    int i = 100; /// 몇번 계산할지에 대한 변수 값
    int size1 = 128; /// A의 wordlen size
    int size2 = 128; /// B의 wordlen size
    /// 위의 세변수만 바꾸어주면 쉽게 연산량을 확인할 수 있음. 밑부분은 수정할 필요 없음.

    int store_i = i;
    srand(time(NULL));
    start = clock(); ///첫번째 측정시작.
    while ((i >= 0)) /// 첫번째 while문에서는 A B 변수를 generate하는 시간과, 해제하는 시간을 측정한다.
    {
        i--;
        bigint_st_ptr A = NULL;
        bigint_st_ptr B = NULL;
        bigint_st_ptr C = NULL;
        generate_random_integer(&A, NON_NEGATIVE, size1);
        generate_random_integer(&B, NON_NEGATIVE, size2);
        delete_bigint(&A);
        delete_bigint(&B);
        delete_bigint(&C);
    }
    end = clock(); ///첫번째 측정완료.
    cpu_time_used1 = ((double)(end - start)) / CLOCKS_PER_SEC;

    i = store_i;
    start = clock(); ///두번째 측정시작
    while ((i >= 0)) /// 두번째 while문에서는 A,B 변수를 generate하고 Addition을 통해 C를 구하는 시간, 그리고 해제하는 시간을 측정한다.
    {
        i--;
        bigint_st_ptr A = NULL;
        bigint_st_ptr B = NULL;
        bigint_st_ptr C = NULL;
        generate_random_integer(&A, NON_NEGATIVE, size1);
        generate_random_integer(&B, NON_NEGATIVE, size2);
        Addition(&A, &B, &C);
        delete_bigint(&A);
        delete_bigint(&B);
        delete_bigint(&C);
    }
    end = clock(); ///두번째 측정완료
    cpu_time_used2 = ((double)(end - start)) / CLOCKS_PER_SEC;
    printf("\nElapsed time with Addition: %lf\n", cpu_time_used2 - cpu_time_used1);
    /// 두번째 구한 시간에서 첫번째 구한 시간을 빼주게 되면, 근사적으로 Addition을 수행하는 시간을 구해줄 수 있다.
    return 0;
}
```

나머지 test_****.c 파일도 위와 같습니다. 총 두가지 함수로 이루어져 있습니다.
(Performance and Verification)

마지막 main.c 함수는 제가 만든 모든 함수를 하나씩 구현할 수 있도록 만들었습니다.
파란색 주석 친 부분이 함수 이름이며, 확인할 함수를 부분적으로 주석 해제
(초록색 부분) 한 후에 컴파일 하시면됩니다.

```
// printf("\n[Is Even or ODD?]\n");///  
// if (is_even_or_od(&A) == EVEN)  
//     printf("EVEN\n");  
// if (is_even_or_od(&A) == ODD)  
//     printf("ODD\n");  
  
// printf("\n[Compare]\n");///  
// if (Compare(&A, &B) == FST_IS_BIG)  
//     printf("Fist is big\n");  
// if (Compare(&A, &B) == SCD_IS_BIG)  
//     printf("Second is big\n");  
// if (Compare(&A, &B) == BOTH_ARE_SAME)  
//     printf("both are same\n");  
  
// printf("\n[Left shift]\n");///  
// Left_Shift(&A, 1);  
// show(A);  
  
// printf("\n[Right shift]\n");///  
// Right_Shift(&A, 16);  
// show(A);  
  
// printf("\n[Reduction]\n");///  
// Reduction(&A, 57);  
// show(A);  
  
// printf("\n[ADD]\n"); ///  
// Addition(&A, &B,&C);  
// show(C);  
  
// printf("\n[SUB]\n"); ///  
// Subtraction(&A, &B,&C);  
// show(C);  
  
// printf("\n[MUL]\n"); ///  
// Multiplication(&A, &B, &C);  
// show(C);
```



한 학기 동안 좋은 수업과 코드 피드백 진심으로 감사드립니다.

많은것을 배울 수 있었습니다. :)