|  |  |
| --- | --- |
| No | Testfile 1 |
| 1 | int sum(int x,int y){ //sum |
| 2 | return x+y; |
| 3 | } |
| 4 | int difference(int x,int y){ |
| 5 | return x-y;//minus |
| 6 | } |
| 7 | int product(int x,int y){ //multi |
| 8 | return (x\*y); |
| 9 | } |
| 10 | int div(int x,int y){ |
| 11 | return x/y; //div |
| 12 | } |
| 13 | int main() { |
| 14 | int a,b,c,d,e,result=0; |
| 15 | a = getint(); |
| 16 | b = getint(); |
| 17 | c = getint(); |
| 18 | d = getint(); |
| 19 | e = getint(); |
| 20 | e = -e+c\*d-a%b+c/(a+b)+sum(a,b)-difference(c,d)\*product(a,b)+div(a,c); |
| 21 | result = result + e\*a; |
| 22 | printf("result is %d.", result); |
| 23 | return 0; |
| 24 | } |
| No | Testfile 2 |
| 1 | int fib(int i){ |
| 2 | if (i == 1) { |
| 3 | return 1; |
| 4 | } |
| 5 | if (i == 2) { |
| 6 | return 2; |
| 7 | } |
| 8 | return fib(i - 1) + fib(i - 2); |
| 9 | } |
| 10 | int main () { |
| 11 | int i = 2,j = 5; |
| 12 | i = getint(); |
| 13 | j = getint(); |
| 14 | //const int a[3][2] = {{1,2},{2,3},{3,4}}; |
| 15 | //const int a[2] = {1,2}; |
| 16 | const int a1 = 1; |
| 17 | const int a2 = 2; |
| 18 | i = (-(i \* j) \* fib(4) + 0 + 2 \* 1 - 1/2) \* 5; |
| 19 | j = 7\*5923/56\*56 - fib(fib(6)) + (1+2-(89/2\*36-53) /1\*6-2\*(45\*56/85-56+35\*56/4-9)); |
| 20 | int k = +-+5; |
| 21 | printf("%d, %d, %d\n", i, j, k); |
| 22 | return 0; |
| 23 | } |
| 24 |  |
| 25 |  |
| No | Testfile 3 |
| 1 | int main () { |
| 2 | /\* dasdfhasdjkhfklsdh |
| 3 | int i = 0; |
| 4 | const char sh = "uzi yyds!!!"; |
| 5 | System.out.println("wu hu qi fei!!!"); |
| 6 | java python |
| 7 | void f() { |
| 8 | void f() { |
| 9 |  |
| 10 | } |
| 11 | function = (msg) => { |
| 12 | // dasdfhasdjkhfklsdh |
| 13 | }; |
| 14 | } |
| 15 | \*/ |
| 16 | printf("hello world\n"); |
| 17 | // printf("hello, world!"); |
| 18 | // /\* \*/dasdfhasdjkh |
| 19 | /\* |
| 20 | //pppp\*/ |
| 21 | printf("end"); |
| 22 | return 0; |
| 23 | } |
| No | Testfile 4 |
| 1 | // test "break" and "continue" in "while stmt" |
| 2 |  |
| 3 | int main() |
| 4 | { |
| 5 | int i, j, k, result; |
| 6 | k = getint(); // j > 0 |
| 7 |  |
| 8 | // sum of k\*k, k\*k-1, ..., k+1, k-1, ..., 2, 1, 0 |
| 9 | i = k \* k; |
| 10 | result = 0; |
| 11 | while (1) |
| 12 | { |
| 13 | if (i < 0) break; |
| 14 | if (i == k) { i = i - 1; continue; } |
| 15 | result = result + i; |
| 16 | i = i - 1; |
| 17 | } |
| 18 | printf("%d\n", result); |
| 19 |  |
| 20 | // sum of k\*k, k\*k-1, ..., k+1 |
| 21 | i = k \* k; |
| 22 | result = 0; |
| 23 | while (1) |
| 24 | { |
| 25 | if (i == k) { break; } |
| 26 | result = result + i; |
| 27 | i = i - 1; |
| 28 | } |
| 29 | printf("%d\n", result); |
| 30 |  |
| 31 | // [1\*1] [1\*2] ... [1\*k] |
| 32 | // sum of [2\*1] [2\*2] ... [2\*k] |
| 33 | // ... ... ... ... |
| 34 | // [k\*1] [k\*2] ... [k\*k] |
| 35 | // [n] = (n > k\*k/2) ? 0 : n |
| 36 | int max = k \* k / 2; |
| 37 | result = 0; |
| 38 | i = 1; |
| 39 | while (1) |
| 40 | { |
| 41 | if ( i > k) break; |
| 42 | j = 1; |
| 43 | while (1) |
| 44 | { |
| 45 | if ( j > k) break; |
| 46 | int tmp; |
| 47 | tmp = i \* j; |
| 48 | if (tmp > max) |
| 49 | { |
| 50 | j = j + 1; |
| 51 | continue; |
| 52 | } |
| 53 | result = result + tmp; |
| 54 | j = j + 1; |
| 55 | } |
| 56 | i = i + 1; |
| 57 | } |
| 58 | printf("%d\n", result); |
| 59 | return 0; |
| 60 | } |
| No | Testfile 5 |
| 1 | // test "RelExp, EqExp" |
| 2 | int maxNumber = 12; |
| 3 | int minNumber = 0; |
| 4 |  |
| 5 | int main() |
| 6 | { |
| 7 | int times, i; |
| 8 | times = getint(); |
| 9 |  |
| 10 | i = 0; |
| 11 | while (i < times) |
| 12 | { |
| 13 | i = i + 1; |
| 14 | int number; |
| 15 | number = getint(); |
| 16 |  |
| 17 | if (number < minNumber) |
| 18 | { |
| 19 | printf("%d: less than minNumber\n\n", number); |
| 20 | continue; |
| 21 | } |
| 22 | if (number > maxNumber) |
| 23 | { |
| 24 | printf("%d: greater than maxNumber\n\n", number); |
| 25 | continue; |
| 26 | } |
| 27 |  |
| 28 | if (number >= minNumber) |
| 29 | { |
| 30 | if (number <= maxNumber) { printf("%d: between minNumber and maxNumber\n", number); } |
| 31 | printf("\n"); |
| 32 | } |
| 33 | } |
| 34 |  |
| 35 |  |
| 36 | return 0; |
| 37 | } |