```c
printf("hello");

int32_t res;

printf("Result = %d", res);

printf("a(%d) * b(%d) = %d", a, b, a * b);

\[ a(5) \times b(3) = 15 \]

INPUT
```
Input

```c
int32_t score;
scanf("%d", &score);
```

Address of `score` in call-by-reference
```c
int8_t N, x, y;
int8_t *p3;
x = 5;
y = 3
p3 = &x;
xp3 = 16

Dereferencing a pointer

scanf("%d
hd", &N) call

scanf(——, int8_t *in) Prototype
*in = "Read from keyboard"
```
```c
uint8_t score1, score2, score3;

uint8_t scores[3];
```

Array is a collection of elements of the same data type

Declaration: `DataType VanName[Size];`

Indices: 0 to `Size-1`

Size: `Size x sizeof(DataType)`

```c
uint32_t Arr2[10]; // Total = 40 bytes
char Name[11]; // = 11 bytes
```
Scores[2] = 75;
l = 1

Scores[l*2]
Scores[2] += 10

Scores[6] = 17; // compiler error

&[Scores[0]] = Scores
Size of an Array

- Explicit
  - `Scores[N]`

- Implicit
  - First element indicates size of array has Sentinel

```
int Scores[6] = {75, 85, 65, 72, 77, -12};
```

Declare + Init

```
int Scores[6] = {51, 75, 65, 68, 72, 77};
```
Subroutine: Input is an Array

Case: Array prefixed by size

```c
void Bonus (int * Arr) {
    int i;
    for (i=1; i<=Arr[0]; i++) {
        Arr[i] += 10;
        i++;
    }
}
```

Call

```c
Bonus (Scores);
```
Subroutine: Input is an Array

Case 2 - Array has sentinel

```c
void Bonus(int* Arr[]) {
    int8_t i;
    for (i = 0; Arr[i] != -1; i++) {
        if (i > 2) Arr[i] += 10;
    }
}
```

```
int16_t Scores[6] = {72, 68, 88, 92, 76, -18};
```

```
Bonus(&Scores[2]);
```
Subroutine : Input is an Array

Case 2 - Array has sentinel

```c
void Bonus (int8_t Arr[]) {
    int8_t i;
    for (i = 0; Arr[i] != -1; i++) {
        Arr[i] += 10;
    }
}
```

```c
int8_t Scores[6] = {72, 68, 88, 82, 76, -18};
Bonus (&Scores[2]);
```
Task: Modify ModProgramming project to validate

✓ Collect all user inputs in an array of size 5 scores

✓ Compute grades for each of the 5 scores and put in array Grades

✓ "Pretty" print scores and grades

\[
\text{Score}[i] \rightarrow \text{Grade}[i]
\]


