Submission 9

```
int32_t high_lvl_0( int32_t i_value ) {
  return i_value;
}
```

returns the input value (a 32 bit integer)

```
uint64_t high_lvl_1( uint64_t ) {
  return 0;
}
```

returns 0 (a 64 bit unsigned integer)

```
int32_t high_lvl_2( int32_t i_option ) {
  int32_t l_result = 0;

  if( i_option < 32 ) {
    l_result = 1;
  }

  return l_result;
}
```

returns 1 if the input value is less than 32, otherwise returns 0 (int32_t)

```
void high_lvl_3( int32_t * i_option,
                 int32_t * o_result ) {
  if( *i_option < 25 ) {
    *o_result = 1;
  }
  else {
    *o_result = 0;
  }
}
```

overrides the value of second input with 1 if the value of the first input is less than 25, otherwise overrides the value of the second input with 0 (both inputs are 32 bit integers)

```
uint32_t high_lvl_4( uint32_t i_x,
                     uint32_t i_y,
                     uint32_t i_z ) {
  uint32_t l_ret = 0;

  if( i_x < i_y && i_x < i_z ) {
    l_ret = 1;
  }
  else if( i_y < i_z ) {
    l_ret = 2;
  }
  else {
    l_ret = 3;
  }

  return l_ret;
}
```

if x is the smallest number, returns 1; if y is the smallest number, returns 2; if z is the smallest number, returns 3;

```
void high_lvl_5( uint32_t   i_nIters,
                 int32_t  * io_value ) {
  for( uint32_t l_i = 0; l_i < i_nIters; l_i++ ) {
    *io_value += 1;
  }
}
```

adds 1 to the value of the second input for the number of times specified by the first input (both inputs are 32 bit integers) ⇒ adds i_nIters to io_value

```
void high_lvl_6( uint64_t   i_nIters,
                 int64_t    i_inc,
                 int64_t  * io_value ) {
  uint64_t l_va = i_nIters;
  do {
    *io_value += i_inc;
    l_va--;
  } while( l_va != 0 );
}
```

adds i_inc to the value of the third input for the number of times specified by the first input (all inputs are 64 bit integers) ⇒ adds i_nIters * i_inc to io_value

```
void high_lvl_7( uint64_t   i_nValues,
                 int64_t  * i_valuesIn,
                 int64_t  * i_valuesOut ) {
  for( uint64_t l_va = 0; l_va < i_nValues; l_va++ ) {
    i_valuesOut[l_va] = i_valuesIn[l_va];
  }
}
```

overwrites the first i_nValues elements of the third input with the first i_nValues elements of the second input (all inputs are 64 bit unsigned integers)