**報告書**

**1.　Bit operator**

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **AND** |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| **1** | **1** | **1** |

|  |  |  |
| --- | --- | --- |
| **A** | **B** | **AND** |
| 0 | 0 | 0 |
| 0 | **1** | **１** |
| **1** | 0 | **１** |
| **1** | **1** | **1** |

**AND(&) 　 OR( | )**

A,B**両方とも** A,B**どちらか**が 　1の場合 　１の場合

|  |  |
| --- | --- |
| **日付** | 2023－06－19 |
| **報告者** | パク・ジュウォン |

・ビット演算子はtrue, false を判明することに優れている。

**２．ポインタ　：アドレス、住所：**

例えば構造体が200byteである場合、より小さいメモリ（4byte）で実行するためにもポインタを使う

|  |
| --- |
| **a = 10** |

|  |
| --- |
| **a: 0x………1** |
| b: 0x………2 |

|  |
| --- |
| **a = 10** |
| ｂ＝… |

**変数aに10を定義**　　　　　　　　　　　**aの住所**　　　　　　　　　　　　　　　**住所値**

**・ポインタ演算子：**

**\*：**演算子を使用してポインタが指すメモリ位置の値を読んだり修正したりすることができる。

**&：**演算子を使用して変数のアドレスを取得できる。

**・ポインタ識別子：**Int\* a : aという変数がint型ポインタだと識別してくれる識別子

**・文字列：char string[0]//** 配列変数はポインタ，配列変数は配列の最初の要素のアドレスを持っている。

string[1] == (\*string +1)

**return**：アドレスにある関数を呼び出して演算がおわったら呼び出したところにReturn（返還）式が終わる。

**２次元配列**

**順番** : 右側から (内部), 左側へ (外部)

int arr[2][3]={ {1,2,3},{4,5,6}};

arr[0][0]=1 arr[0][1]=2 arr[0][2]=3

arr[1][0]=4 arr[1][1]=5 arr[1][2]=6

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| **arr[0][0]=1** | | | | **arr[0][1]=2** | | | | **arr[0][2]=3** | | | | **arr[1][0]=4** | | | | **arr[1][1]=5** | | | | **arr[1][2]=6** | | | |