Reference (no)

- v C je možné předávání parametrů pouze hodnotou (výjimkou je pouze pole)
- v případě, že v C potřebujeme další „odkaz“ na již existující proměnnou (v tom samém bloku, nebo předat jako parametr funkci), řešíme pomocí ukazatele – lze i v C++ ukazatel je lokální hodnota (adresa do paměti)

- nakreslete umístění proměnných z následujícího příkladu v paměti

```
// dvě proměnné předané hodnotou, první typu int,
// druhá typu ukazatel
// obě jsou definice lokálních proměnných s inicializací
void funkce(int aParam, int *aProm)
{
 int *prom; //  neinicializovaná proměnná
 prom = &aParam; // druhý parametr pro práci s aParam
 *aProm = 3; // práce s předanou proměnnou (zde bbb)

}

int bbb,a = 3;
funkce (a, &bbb); // předání – inicializace parametrů
```

Reference (no)

- nový datový typ v C++
- reference – odkaz (alias, přezdívka, nové jméno pro stávající proměnnou)
- zápis proměnné typu reference je **Typ&** a musí být inicializována ihned při definici
- obdobně jako předávání ukazatelem, reference „šetří čas“ a paměť při předávání parametrů do a z funkcí

```
T tt, &ref=tt; //  definice reference povinně s inicializací
extern T &ref; //  deklarace reference

Typ& pom = p1.p2.p3.p4; // zjednodušení zápisu pro přístup
```

Varianty použití operátoru &:

```
int aa, bb, cc, *pc;
cc = aa & bb; // & značí operátor logické and bit po bitu
pc = &cc;     // & značí operátor získání adresy prvku
int &rx = aa; // & značí že je definována proměnná jako reference
// reference je to v definici proměnné na levé straně =
int *pa = &aa;//=získání adresy, je v definici, ale napravo =
```

Reference (no)

Napište funkce Real, která má parametr typu reference double a vrací double. Funkce nastavuje předanou hodnotu na 4. Ukažte volání této funkce a popište co se děje s proměnnými a jejich hodnotami

Napište funkce Real, která má parametr typu reference double a vrací double. Funkce nastavuje předanou hodnotu na 4. Ukažte volání této funkce a popište co se děje s proměnnými a jejich hodnotami

```
double Real(double &aVa)//předání proměnné referencí do funkce
{aVa = 4;
return aVa;}

double ab; Real(ab); // způsob volání

// aVa je nové jméno pro volající proměnnou =
// dvě jména/proměnné se dělí o společné místo v paměti.
// Přístup k hodnotě ab i aVa směřuje do téhož místa v paměti.
// Při přiřazení do aVa dojde i ke změně původní proměnné ab.
// Reference umožní změny vně,úspora proti volání hodnotou.
// sndnější zápis při psaní těla funkce
```

- ”splývá” předávání i práce při předání hodnotou a referencí (až na prototyp stejné)
- práce s referencí = práce s původní odkazovanou proměnnou
- nelze reference na referenci, na bitová pole,
- nelze pole referencí, ukazatel na referenci
- je možné vracení parametrů odkazem (je možné vrátit pouze parametry (proč?))

Reference (no)

Napište funkci, která má dva parametry. Jeden předávaný referencí, druhý pomocí ukazatele. Funkce vrátí prvek z větší hodnotou pomocí reference. Vysvětlete způsob předávání proměnných („jak to vypadá v paměti“ během programu). Ukažte a vysvětlete použití funkce „na levo“ od rovná se funkce() = proměnná;

Funkce má dva parametry. Jeden předávaný referencí, druhý ukazatelem. Funkce vrátí prvek z větší hodnotou pomocí reference. „Jak to vypadá v paměti“ během programu?

```
double& Funkce(double &p1, double *p2)
{  double aa;
   p1 = 3.14; // pracujem jako s proměnnou
   // return aa; // nelze - aa neexistuje vně
   if (p1 > *p2)
      return p1; // lze – existuje vně
   else
      return *p2; // lze – existuje vně
   //vrací se ”hodnota”,referenci udělá překladač
   // odkazujem se na proměnnou vně Funkce
}

double bb,cc,dd ; //ukázka volání
dd = Funkce (bb,&cc); // návratem funkce je
// odkaz na proměnnou, s ní se dále pracuje
Funkce(bb,&cc) = dd; // vrací odkaz
```

U ukazatele je jasně vidět z přístupu, že je to ukazatel (& a *)
U reference je předávání a práce jako u hodnoty, liší se pouze v hlavičce