& référence

① `int i(3); int& ref = i;`

② passage par réf. `double f(vector<int>& tab, int a);` → MODIF possibles

③ `z = f(v, 32);`

boucle qui modifie :

```
for(auto& x : tab) { x = ... ; }
```

`&x` prend l'adresse de x


```
double* ptr(nullptr); ... ; ptr = &x;
```

*

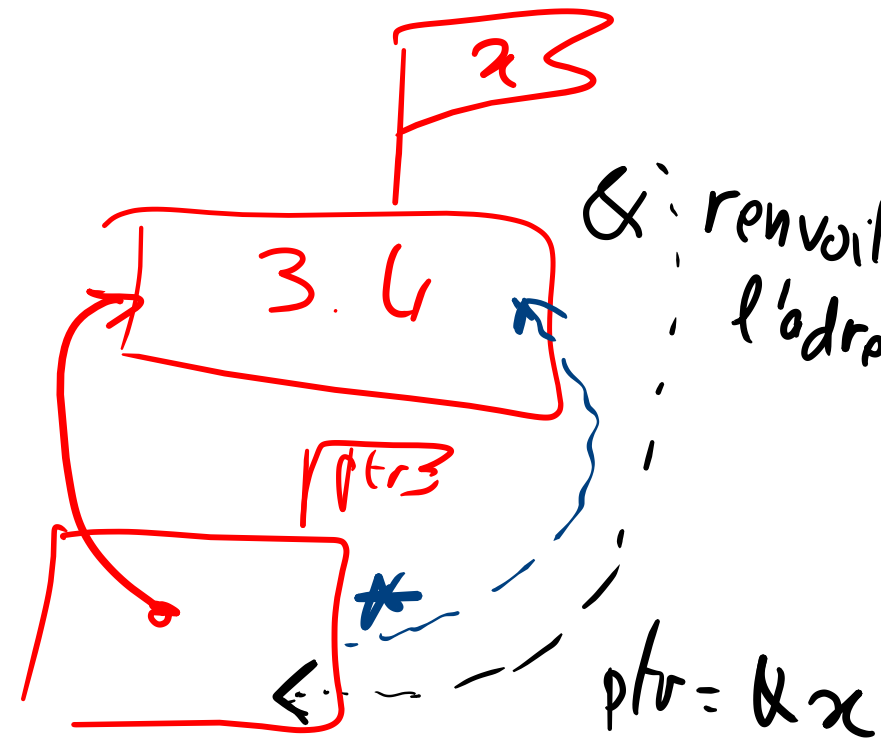
(1) multiplication : a * b

(2) sert à déclarer un ptr : double * ptr;

(3) accès à valeur pointée

double x (3.4);

double* ptr (& x);



*ptr

de type double et vaut 3.4

Question 1 – Florilège [24 points]

① **[2 points]** Si un nombre entier x s'écrit `0010000000000000` en binaire sur 16 bits, comment s'écrit $\log_2(x)$ en binaire sur 16 bits ?

Réponse :

② **[1 point]** Sur une machine sur laquelle les `size_t` sont sur 32 bits, quelle est la représentation binaire de la valeur de la variable `s` suivante :

```
const size_t s(-1);
```

Réponse :

③ **[2.5 points]** Qu'affiche l'extrait de code suivant :

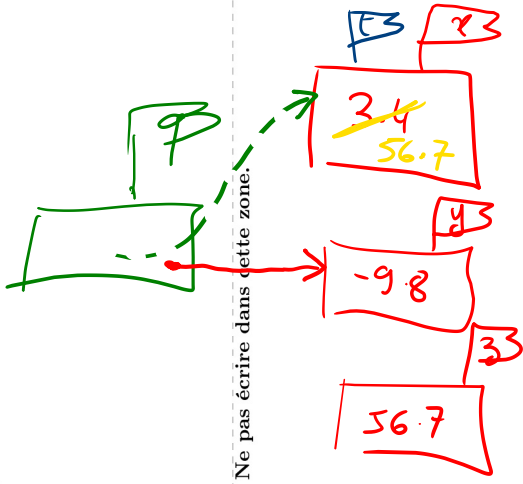
```
double x(3.4);
double y(-9.8);
double z(56.7);

double& t(x);

double* q(&x);

t = z;
q = &y;

cout << "x = " << x << ", y = " << y << ", z = " << z
     << ", t = " << t << ", q ==> " << *q << endl;
```



Réponse : `x =` , `y =` , `z =` , `t =` , `q ==>`

④ **[3 points]** En utilisant RSA, vous souhaitez protéger une information que vous envoyez à un ami dont la clé publique est $(25, 95)$.

Sachant que votre clé publique est $(29, 51)$ et que votre clé privée est 21, vous lui envoyez 23.
Votre ami, dont la clé privée est 49, déchiffre le message qu'il a reçu et trouve 41.

Que pouvez vous dire de la transmission ? Justifiez pleinement votre réponse.

Note : quelques valeurs de $a^b \mod c$ (classées par a, b, c) :

$23^{21} = 23 \mod 32$	$23^{21} = 41 \mod 51$	$23^{25} = 23 \mod 72$	$23^{25} = 63 \mod 95$
$23^{29} = 23 \mod 32$	$23^{29} = 44 \mod 51$	$23^{49} = 23 \mod 72$	$23^{49} = 28 \mod 95$
$41^{21} = 9 \mod 32$	$41^{21} = 11 \mod 51$	$41^{25} = 41 \mod 72$	$41^{25} = 21 \mod 95$
$41^{29} = 9 \mod 32$	$41^{29} = 23 \mod 51$	$41^{49} = 41 \mod 72$	$41^{49} = 71 \mod 95$

Réponse :

Ne pas écrire dans cette zone.

Question 5 – Le compte est bon [27 points]

On s'intéresse ici à écrire des *parties* d'un programme C++ permettant de résoudre le problème du « compte est bon » : à partir de 6 nombres entiers donnés et d'un résultat désiré (aussi entier), essayer de combiner un sous-ensemble quelconque de ces 6 nombres, pris chacun qu'une seule fois, pour obtenir le résultat.

Par exemple, peut-on trouver le résultat 329 à partir des nombres $(2, 5, 5, 8, 25, 50)$?
La réponse est « oui » : $(5 + 2) \times (50 - (8 - 5)) = 329$.

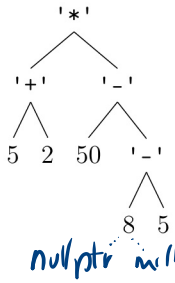
① **[3 points]** Dans un premier temps, on s'intéresse à représenter une suite de calculs arithmétiques.
Pour cela, un *calcul* est défini comme :

- une opération ($+$, $-$, $\times$ ou $/$) (un `char` suffira en C++) ;
- un résultat (entier positif) ;
- un calcul, opérande de droite ;
- un calcul, opérande de gauche.

Par exemple, le calcul $(5 + 2) \times (50 - (8 - 5))$ peut se représenter comme :

- opération : `'*'` ;
- résultat : 329 ;
- calcul de gauche : « $(5 + 2)$ » ;
- calcul de droite : « $(50 - (8 - 5))$ ».

Évidemment, les deux derniers éléments sont également des *calculs*, chacun avec leurs éléments respectifs. Le calcul « $(5 + 2) \times (50 - (8 - 5))$ » complet pourrait donc se représenter (sans les résultats) comme ci-contre :



Proposez un type `Calcul` pour représenter en C++ les calculs tels que définis ci-dessus.

Réponse :

② **[3 points]** Comment représentez-vous les nombres de base utilisés dans les calculs (comme p.ex. 8, 5, 50 ou 2 dans le calcul précédent) ?

Définissez p.ex. en C++ la variable représentant le nombre 50 du calcul précédent.

Réponse :

Ne pas écrire dans cette zone.

f(ifstream & flot)

double x(3.14);

const double * const ptr (&x);

ne peut pas modif la valeur (*ptr)

ne peut pas modifier l'adresse

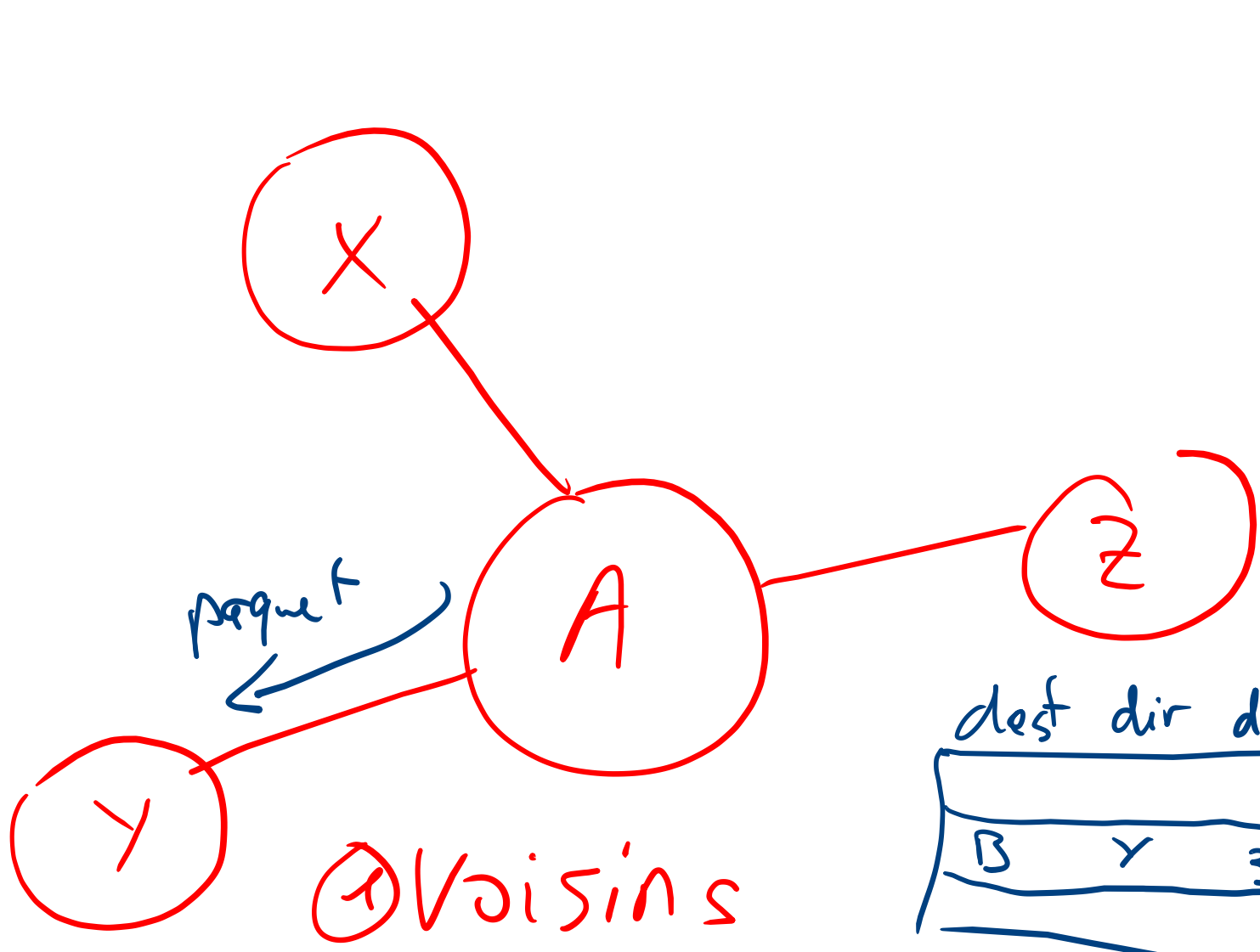
pgm.cc

```
ifstream f("fichier.txt")
int a;
f >> a;
```

disque

42

"fichier.txt"



① Voisins

table de A

dest	dir	dist
B	Y	3

② quelle route ? → aucune idée
juste choisir le voisin "qui va bien"
dans la table