

Devoir maison info

question 1

1.

correcte

affichage : 42_42

2.

incorrecte:

le "a" dans le printf est une adresse, hors dans le guillemet du printf nous avons un %d qui représente un entier

3.

incorrecte

un pointeur "a" est déclaré, hors elle n'est pas initialisée à une adresse précise, et on effectue une affectation sur le pointeur mais ce pointeur pointe vers NULL

4.

incorrecte

"*a" est de type int, "&n" est une adresse, *a = &n => incompatible

5.

incorrecte

"a" est une adresse, 42 est un entier, de type incompatible

6.

correcte

affichage: 42_42

7.

correcte

affichage: 42_42

8.

incorrecte

"n" n'est pas initialisé dans la fonction

9.

correcte

(ici le programme va compiler car on a allouer l'espace mémoire pour le pointeur "a" dans le tas, et comme elle n'est pas initialiser par défaut elle vaut 0)

affichage: 42_0

10.

correcte

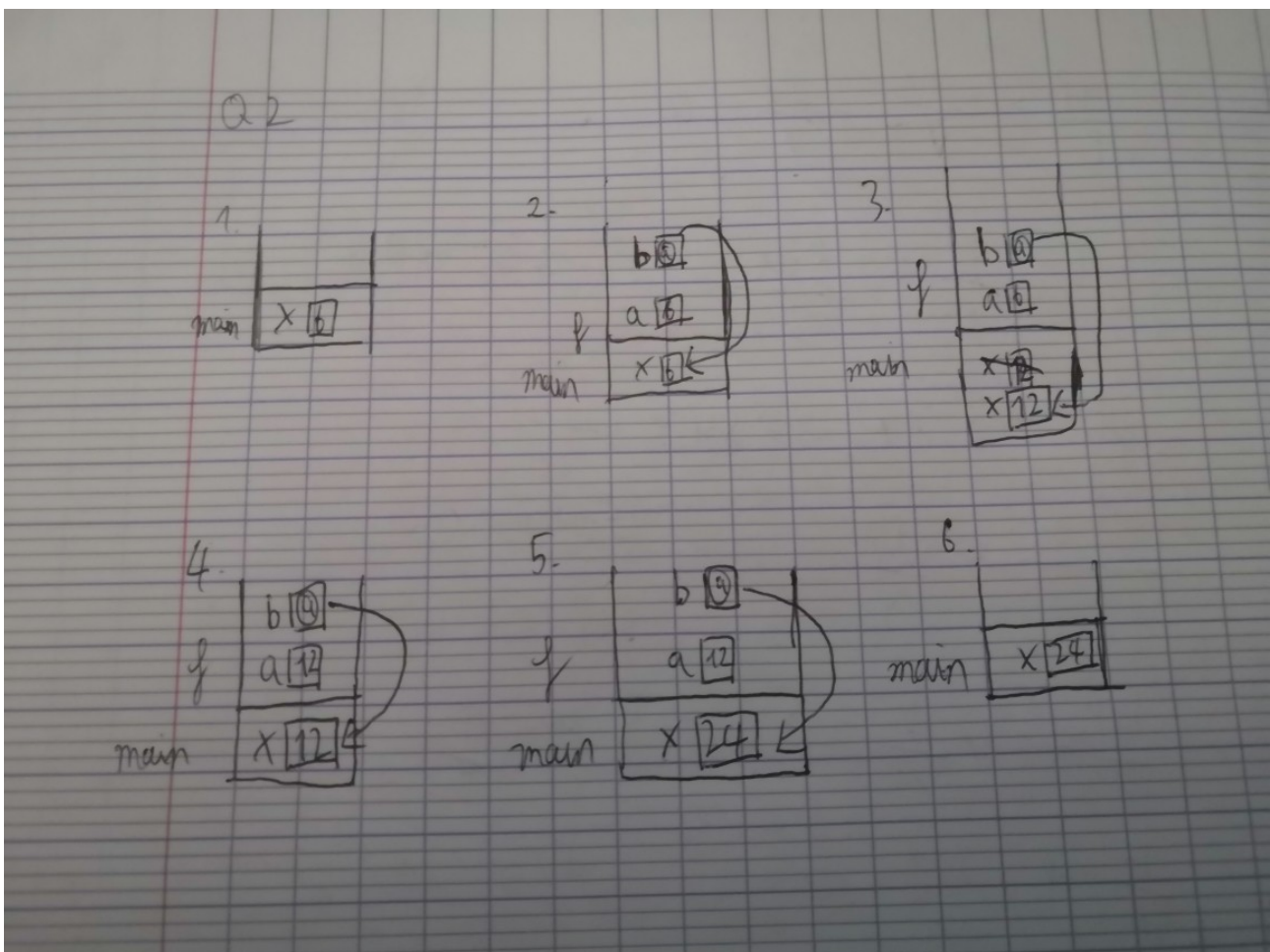
affichage: 42_42

11.

correcte

affichage : 42_42

question 2



question 3

```
#include<stdlib.h>
#include<stdio.h>
```

```

int points(int e1, int e2)
{
    int ptn, dif;
    dif = e1-e2;
    if(dif >= 4)
    {
        ptn = 0;
    }
    else
    {
        if(dif==3)
        {
            ptn =15;
        }
        else
        {
            if(dif==2)
            {
                ptn = 20;
            }
            else
            {
                if(dif == 1)
                {
                    ptn = 30;
                }
                else
                {
                    if(dif == 0)
                    {
                        ptn = 60;
                    }
                    else
                    {
                        if(dif == -1)
                        {
                            ptn = 90;
                        }

                        if(dif<=-2)
                        {
                            ptn = 120;
                        }
                    }
                }
            }
        }
    }

    printf("%d\n", ptn);
    return ptn;
}

```

tableau

```
#include<stdio.h>
#include<stdlib.h>
#define K 5
#define N 2
```

// question 4

```
int val_pos(int n, int t[],int len)
{
    int i,k=0;
    for(i=0; i<len; i++)
    {
        if(t[i] > 0)
        {
            k += 1;
        }
    }
    if(k >= n)
    {
        return 1;
    }
    return 0;
}
```

// question 5

//premier occurrence

```
int indice_min(int t[], int len)
{
    int k = t[0];
    int i;
    for(i=0; i<len; i++)
    {
        if(k > t[i])
        {
            k = t[i];
        }
    }
    int j;
    for(j=0;j<len;j++)
    {
        if(t[j] == k)
        {
            printf("%d\n",j);
            return j;
        }
    }
}
```

```
    }  
}
```

//question 6

```
void replace_min_if_sup(int n, int t[], int len)  
{  
    int i = indice_min(t,len);  
    if(n>t[i])  
    {  
        t[i] = n;  
    }  
}
```

//question 7

```
int *t_best(int n , int t[], int len)  
{  
    int k = t[0];  
    int *tab = malloc(n * sizeof(int));  
    int i;  
    for(i=0; i < len; i++)  
    // le trouver un élément qui est supérieur au plus grand élément du tableau  
    {  
        if(k< t[i])  
        {  
            k = t[i];  
        }  
    }  
    k += 1; // forcément exclus du tableau et supérieur au plus grand elem du tableau  
    int j;  
    for(j = 0; j < (len - n); j++)  
    {  
        replace_min_if_sup(k,t,len);  
    }  
  
    int p, z = 0;  
    for(p=0; p<len ; p++)  
    {  
        if(t[p] != k)  
        {  
            *(tab+z) = t[p];  
            z +=1;  
        }  
    }  
  
    return tab;  
}
```

tableau et adresse

//question 8

```
void echange(int t1[], int t2[], int len)
{
    int i;
    for(i=0; i< len; i++)
    {
        if(t1[i]<t2[i])
        {
            int tmp;
            tmp = t1[i];
            t1[i] = t2[i];
            t2[i] = tmp;
        }
    }
}
```

//question 9

en executant echange(t3,&t3[6],6); nous comparons la premiere partie du tableau, du 0 eme element au 5 eme element ,et la deuxieme partie du tableau, du 6 eme element au 11eme element. Car &t3[6] correspond a l'adresse du 6 eme element. et ensuite la fonction va echanger les valeurs de t3, si t3[i]<t3[i+6].

t3 donc vaut : { 7, 4, 9, 4, 6, 7 ,6, 3, 8, 3, 5, 5 }.