The exam answer.

Exercice #1

Q1 : Execution

a. Values of Variables **1,25pts**

N	a	X	y	d	c	Observations
22327	2	22327	/		0	befor loop
		2232	2232	7	0	1st iteration
		223	223	2	1	2nd iteration
		22	22	3	1	3rd iteration
		2	2	2	2	4th iteration
		0	0	2	3	5th iteration

b. The execution screen   **0,25pts**

```
enter a positive integer:    22327
enter an integer ≥ 0 and ≤9:  2
result : 3
```

Q2: The algorithm calculates the number of occurrence of a digit in a positive integer.   **0,5pts**

Q3:
```
#include <stdio.h>
int main()
{   int a,b,c,n,x,y,d;
    printf("enter a positive integer : ");
    scanf("%d",&n);
    printf("enter an integer ≥ 0 and ≤9:");
    scanf("%d",&a);
    c=0;
    x=n;
    while(x>0)
    {   y=x/10;
        d=x-y*10;
        if(d==a)
           c++;
        x=y;
    }
    printf ("result : %d ",c);
}
```

2,5pts

(-0.25 for each fault)

Q4 : $d \leftarrow x- y*10 \iff d \leftarrow x \bmod 10$ **0,5pts**

Q5 :   *write (" enter an integer ≥ 0 and ≤9: ")*
       do
          read(a)
       while (a<O or a>9)   **0,5pts**

Q6*:* we **cannot** use the for loop because we **do not know in advance the number of iterations** of the loop.   **0,5pts**

Q7

4pts

```
Algorithm exam
Var
     a, c, n , x, y, d :integer
T[10] : array of integer;                          0,5pts
Begin
  write ( " enter a positive integer:  " )
  read(n )
  write ( " enter an integer ≥ 0 and ≤9: " )
   for i ←0 to 9 do
      T[i] ←0
   End for                                         01pts
  x←n
  while (x>0) do
     d ← x mod 10
```

```
            T[d] ← T[d]+1
            x ← x div 10
          end while
          write (" result : ")
          for i ← 0 to 9 do
              write ( i , ": " , T[i], "times")           1pts
           End for
        End
```
                                                        1,5pts

Exercice #2

Algorithm matrix
Var
  i, j ,R,C:**integer**
    A[10][10]: **array of integer**   **0,5pts**
Begin
    write("enter number of rows and number of columns : ")
    read(R,C)   **0,5pts**
    for i←0 **to** R-1 **do**
      for j←0 **to** C-1 **do**
      A[i][j] ←(i+1)*(j+1)   **2pts**
       endfor
    endfor
    write("the resulting matrix :")
  for i←0 to R-1 **do**   **1pts**
      for j←0 **to** C-1 **do**
        write (t[i][j])
      endfor
    endfor
End

Exercise #3

```c
#include<stdio.h>
typedef  struct {
      float  a,b,c;} equation ;                         1pts
int main()
    {   equation T[100];                                0,5pts
        int i , n , nb;
        float delta ;                                   0,25pts
        printf (" give the number of equations");
        scanf("%d",&n);                                 0,25pts
        printf ("enter the coefficients  of each equation  : \n");
        for(i=0;i<n;i++)                                01pts
        {       printf ("equation  %d \n", i+1);
                scanf("%f%f %f ",&T[i].a, &T[i].b, &T[i].c);
        }
        nb=0;                                           0,5pts
        for(i=0;i<n;i++)
          {      delta = T[i].b* T[i].b-4* T[i].a* T[i].c;    1,5pts
                 if (delta >=0)
                    nb++;                                0,5pts
          }
        printf("The number of equations that admit at least one solution: %d equations",nb);  0,5pts
}
```