11, December 2024 **Computer science MIDTERM - S1** *Duration : 50min*

Full Name : ***Section/Group :***

Exercice 1 : (2pts)

A bank is open all days from **8:00** to **12:00** and from **14:00** to **17:00**, except on thursday afternoon and all day friday.

Let **h** be a time **(in hour)** assumed to be an integer between **0** and **23**. And let **d** be a day assumed to be an integer between **1** to **7** (**1** for Saturday, **2** for Sunday, etc.).

Write a program that reads the time **h**, and the day **d**, then displays whether the bank is open or closed.

Examples :

d=6 h=14 : the bank is **closed**
d=2 h=14 : the bank is **open**
d=4 h=12 : the bank is **closed**
d=3 h=8 : the bank is **open**
d=5 h=17 : the bank is **closed**

Solution :

```
#include <stdio.h>
int main() {     int  d, h;
```

.....

.....

.....

.....

.....

.....

.....

Exercise 2 : (2.5pts)

```
float  S;   int N, y, i;
 (1) y=N=123;
 (2) S =0;
 (3) i=1;
     while (y != 0)   {
 (4) S = S + (y%10)*pow(10,-i);
 (5) i++;
 (6) y = y / 10;
     }
 (7) S=S*pow(10, i-1);
     printf("S=%f", S);
```

1) Unroll this program
2) What does the final value of S represent?

.....

.....

.....

.....

.....

1) Unrolling :

Action N°	N	y	i	S
(1)				

Exercise 3 : (5.5pts)

Let **A** be an array of **n** integers (**n**<=50); Write a program that :
1) Fill in the array **A**.
2) Calculate the average of odd elements of **A**

Example : **A**

7	6	1	0	-1	8	3

The average is = (7+1-1+3)/4=2.5

3) Swap between each two adjacent elements of **A** (if the size of **A** is odd, the last element doesn't change)

Example : **A** | 7 | 6 | 1 | 0 | -1 | 8 | 3 | ⇒ **A** | 6 | 7 | 0 | 1 | 8 | -1 | 3 |

4) Construct another array **B** that contains the even elements of **A**.

Example : **A** | 6 | 7 | 0 | 1 | 8 | -1 | 3 | ⇒ **B** | 6 | 0 | 8 |

5) Display the elements of **B** in the following format :

B[1]=6 B[2]=0 B[3]=8

Solution :

```
#include <stdio.h>
int main() {
```

Bonus : (1pt)

Let **n**, **p** two integers (**n**>0 and **p**>0).

The following program allows to calculate and display the sum **S** of the **n** first integers starting from **p. Without using loops,** write an action that calculates the value of **S.**

Example : if p=**3**, n=**5**

S= **3**+4+5+6+7=25

(**5** numbers)

Solution :

```
#include <stdio.h>
int main() {
int n, p, S;
do  {
   printf("Enter two integers n, p, with : n>0 and p>0 :");
   scanf("%d%d", &n, &p);
} while (! (p>0 && n>0));

S=.....

.....

.....

printf("Sum=%d", S);
return  0;   }
```