

Surname (readable)..... Name (readable).....

Matr..... Signature.....

Q1	Q2	TOT

NOTES

It is forbidden to refer to texts or notes of any kind as well as interact with their neighbors. Anyone found in possession of documents relating to the course, although not directly relevant to the subject of the examination will cancel the test. It is not allowed to leave during the first half hour, the task must still be returned, even if it is withdrawn. The presence of the writing (not delivered) implies the renunciation of any previous ratings.

Question 1 (12 points)

Given the following task set, you are asked to draw the Gantt diagram of the schedule, obtained by applying Earliest Deadline First (EDF) algorithm, once verified the schedulability of the task set.

NOTE: In case of tasks with the same priority, pick the one with the shortest relative deadline.

Task	WCET (C)	Deadline (D)
0	1	4
1	3	12
2	2	8
3	1	6

Do the same with for the following taskset, by using Deadline Monotonic (DM)

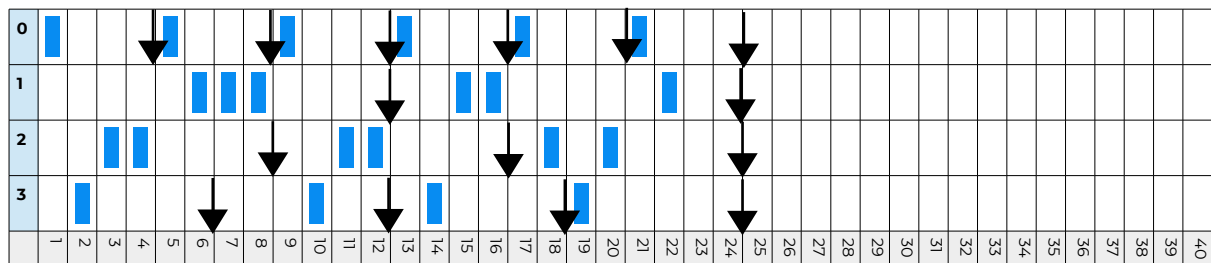
Task	WCET (C)	Deadline (D)	Period (T)
0	1	4	5
1	3	9	10
2	2	8	10

1) Earliest Deadline First (EDF)

Task	WCET (C)	Deadline (D)
0	1	4
1	3	12
2	2	8
3	1	6

$$U = 1/4 + 3/12 + 2/8 + 1/6 = 0.9167 < 1 \rightarrow \text{OK}$$

Schedule



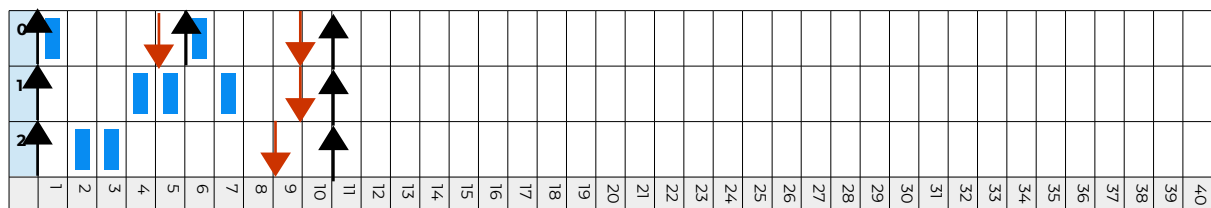
2) Deadline Monotonic (DM)

Task	WCET (C)	Deadline (D)	Period (T)
0	1	4	5
1	3	9	10
2	2	8	10

$$U_{\max} = n(2^{1/n} - 1) = 0.7797$$

$$U = 1/4 + 2/9 + 2/8 \rightarrow 0.8333 > 0.7797 \rightarrow \text{No guarantees}$$

Schedule



Question 2 (11 points)

A 32 bit ARM CPU is connected to 32KByte of FLASH starting from address 0x2000 and 4KByte of RAM mapped immediately after the FLASH with no gap.

Using the provided template and by filling in the blanks, write a linker script ONLY (no startup assembler script, just the linker script). The linker script shall support the four basic sections required to support C programs. Disregard any alignment requirements (.= ALIGN statements).

```
ENTRY(Reset_Handler)
MEMORY {
    flash(rx) : ORIGIN = _____, LENGTH = _____
    ram(wx)    : ORIGIN = _____, LENGTH = _____
}
_stack_top = _____ ;
SECTIONS
{
    . = 0;
    .text : {
        KEEP(*(.isr_vector))

    } > _____
    _etext = .;
    .data : {
        _data = .;

        _edata = .;
    } > _____
    _bss_start = .;
    .bss : {

    } > _____
    _end = .;
}
```

Solution:

```
ENTRY(Reset_Handler)
MEMORY {
    flash(rx) : ORIGIN = 0x2000, LENGTH = 32K
    ram(wx)    : ORIGIN = 0xa000, LENGTH = 4K
}
_stack_top = 0xb000 ;
SECTIONS
{
    . = 0;
    .text : {
        KEEP(*(.isr_vector))
        *(.text)
        *(.rodata)
    } > flash
    _etext = .;
    .data : {
        _data = .;
        *(.data)
        _edata = .;
    } > ram AT > flash
```

```
    _bss_start = .;  
    .bss : {  
        *(.bss)  
    } > ram  
    _end = .;  
}
```