

Questão 1 (10 pontos) – Sabe-se as variáveis X e Y, contém respectivamente os valores “AA” e “56” (ambos em hexadecimal). Considerando estas variáveis são do tipo *unsigned char*, qual o valor resultante da expressão: $Z = X \mid Y$?

- a) FEh
- b) EBh
- c) 21h
- d) VERDADEIRO (TRUE)
- e) FALSO (FALSO)

Questão 2 (10 pontos) - Considere que você deve declarar uma variável que irá armazenar a temperatura lida por um sensor. Sabe-se que este sensor irá monitorar a temperatura de uma câmara climatizada e que esta pode variar de -75°C a +120°C. O tipo de dado recomendado para a variável que irá armazenar as leituras de temperatura é:

- a) Int
- b) Signed Int
- c) Unsigned Int
- d) Unsigned Char
- e) Char

Questão 3 (10 pontos) - Em linguagem C é muito comum a utilização do comando “const” para se definir uma constante. Este possibilita a atribuição de um nome a um valor numérico constante, o que facilita a interpretação do código.

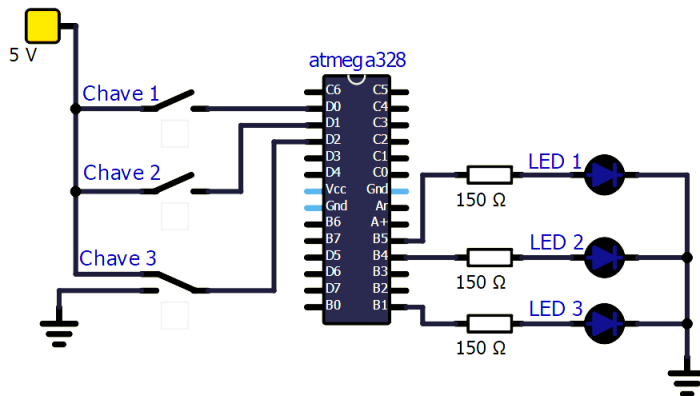
Com relação a este comando, são feitas as seguintes afirmações:

- I. O valor atribuído em um comando “const” pode ser alterado ao longo do programa, porém fora do programa principal.
- II. O valor atribuído em um comando “const” pode ser alterado ao longo do programa, porém apenas dentro do programa principal.
- III. Após a execução do comando “const int IDADE=18”, o termo “IDADE” será sempre substituído pelo valor numérico “18”.
- IV. É comum a definição de várias constantes em um único comando, como: const int PINO1=0, PINO2=1, PINO3=1;

Pode-se considerar como **verdadeira(s)** a(s) seguinte(s) afirmação(ões):

- a) Apenas III.
- b) Apenas III e IV.
- c) Apenas I, II e III.
- d) Apenas II, III e IV.
- e) Apenas IV.

Questão 4 (10 pontos) - Dado o circuito abaixo, no qual o microcontrolador está programado com o código ao lado.



```
int main() {
    DDRD = 0;
    PORTD = 0xFF;
    DDRB = 0b11111111;
    PORTB = 0;
    while(1) {
        if (PIND & (1 << PD0)) {
            PORTB |= (1 << PB5);
        } else {
            PORTB &= ~(1 << PB5);
        }
        if (PIND & (1 << PD1)) {
            PORTB |= (1 << PB4);
        } else {
            PORTB &= ~(1 << PB4);
        }
        if (PIND & (1 << PD2)) {
            PORTB |= (1 << PB1);
        } else {
            PORTB &= ~(1 << PB1);
        }
    }
}
```

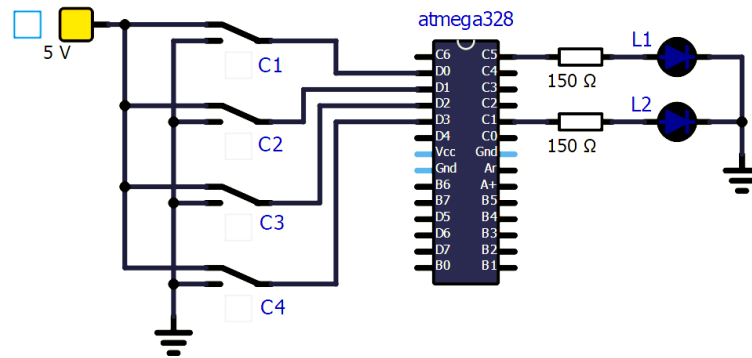
Considerando as informações anteriores, é correto afirmar que:

- I. Caso as chaves estejam nas posições mostradas na figura, todos os LEDs estarão acesos
- II. As chaves 1 e 2 não exercem nenhuma função pois não alteram o estado de nenhum registrador.
- III. Para que os LED 1 e 2 sejam apagados com o fechamento das chaves 1 e 2, é necessário conectar as chaves ao GND e alterar incluir a condição de inversão nos dois primeiros "If" no código.

Pode-se considerar como **verdadeira(s)** a(s) seguinte(s) afirmação(ões):

- a) Apenas I e II.
- b) Apenas I e III.
- c) Apenas II e III
- d) Nenhuma.
- e) Todas

Questão 5 (10 pontos) – Considere que você deve programar o ATMEGA328 apresentado no circuito da figura seguinte. As chaves (C1, C2, C3 e C4) conectadas às entradas do portal D serão utilizadas para controlar os LEDs (L1 e L2).



Para que isto seja possível é necessário que o estado lógico nas entradas PD0, PD1, PD2 e PD3 sejam constantemente monitorados, de forma a fazer com que o estado dos LEDs seja alterado com base no nível lógico alto nas entradas. Todos os resistores de *pull-up* no portal D foram desabilitados. A estrutura de decisão mais adequada para o monitoramento de PD2, por exemplo, é a seguinte:

- a) If ((PIND | PD2) == 0)
- b) If ((PIND && PD2) != 0)
- c) if (PIND & (1<<PD2))
- d) if (!(PIND & (1<<PD2)))
- e) Nenhuma das alternativas anteriores está correta.

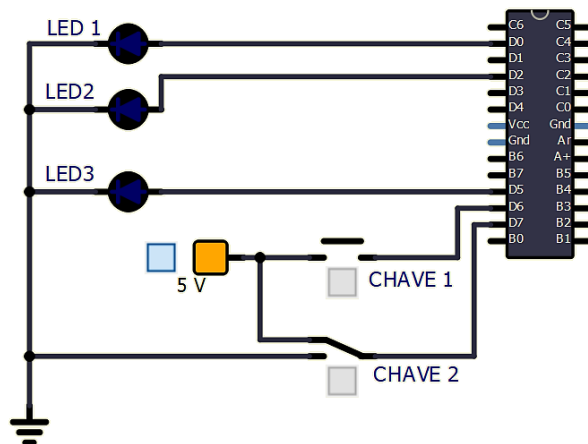
Questão 6 (10 pontos) - Em desenvolvimento de firmware, em geral, é muito comum a utilização do recurso de interrupção. Este possibilita o atendimento a uma situação de forma imediata, no momento em que é requisitada. Com relação as interrupções, são feitas as seguintes afirmações:

- I. As interrupções fixas são assim chamadas por possuírem um endereço fixo de memória. São normalmente interrupções internas ligadas aos periféricos.
- II. As interrupções externas podem ser solicitadas de qualquer pino de GPIO.
- III. As interrupções demandam a ativação do serviço global de interrupções (sei()).

Pode-se considerar como **verdadeira(s)** a(s) seguinte(s) afirmação(ões):

- a) Apenas I e II.
- b) Apenas I e III.
- c) Apenas II e III
- d) Nenhuma.
- e) Todas

Questão 7 (10 pontos) - Considere o seguinte circuito com duas chaves e três LEDs ligados à GPIO do ATMEGA328 e o estado atual dos registradores mostrado à direita do circuito.

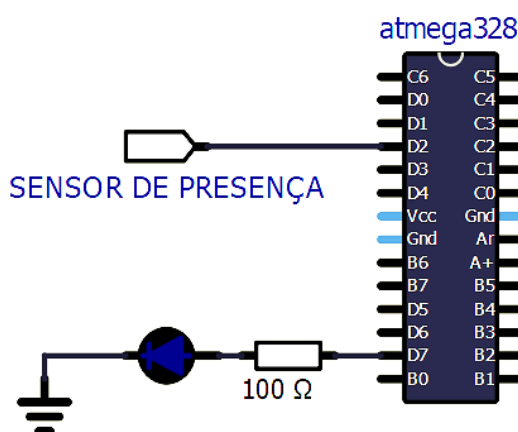


DDRD	0	0	0	0	0	1	0	1
PORTD	0	1	0	1	1	1	1	0

Com base nisto, qual o estado atual do registrador PIND?

- a) 0 0 0 0 0 1 0 1
- b) 0 0 0 0 0 1 0 1
- c) 0 0 0 0 0 1 0 1
- d) 1 1 0 1 1 * 1 *
- e) 0 0 0 0 0 1 0 1

Questão 8 (10 pontos) - Considere um sensor de presença conectado ao ATMEGA328, capaz de detectar a presença de objetos próximos à sua face sensora. O mesmo fornece nível lógico alto ao ATMEGA sempre que detecta a presença de um objeto e nível lógico baixo quando não detecta a presença do objeto. O microcontrolador está programado com o firmware à seguir:



```
#define led PD7
ISR(INT0_vect) {
    PORTD|=(1<<led);
    _delay_ms(2000);
    PORTD&=~(1<<led);
}

int main() {
    DDRD|=(1<<led);
    EICRA|=(1<<ISC01); //borda de descida
    EIMSK |= (1<<INT0);
    sei();
    while (1) {
    }
}
```

Com base nas informações apresentadas, pode-se afirmar que:

- a) O LED se apagará por 2 segundos sempre que o sensor detectar a presença de um alvo.
- b) O LED se acenderá por 2 segundos sempre que o sensor parar de detectar a presença de um alvo.
- c) O LED se apagará por 2 segundos sempre que o sensor parar de detectar a presença de um alvo.
- d) O LED se acenderá por 2 segundos sempre que o sensor detectar a presença de um alvo.
- e) Nenhuma das outras alternativas

Questão 9 (10 pontos) - Considere que você deseje alterar o firmware do ATMEGA328 da questão anterior afim de que não seja utilizada nenhuma interrupção. O LED deverá acender sempre que um objeto for detectado. Para isto, você o definiu como PD7, configurou o sensor de presença como entrada e habilitou o *pull-up* neste pino. Assim, o loop infinito deverá ser programado com a seguinte lógica:

a)

```
while (1) {  
    if (PIND&(1<<led)) {  
        PORTD|=(1<<led);  
        _delay_ms(2000);  
        PORTD&=~(1<<led);  
    }  
}
```

b)

```
while (1) {  
    if (!(PIND&(1<<led))) {  
        PORTD|=(1<<led);  
        _delay_ms(2000);  
        PORTD&=~(1<<led);  
    }  
}
```

c)

```
while (1) {  
    if (PIND&&(1<<led)) {  
        PORTD|=(1<<led);  
        _delay_ms(2000);  
        PORTD&=~(1<<led);  
    }  
}
```

d)

```
while (1) {  
    if (!(PIND&&(1<<led))) {  
        PORTD|=(1<<led);  
        _delay_ms(2000);  
        PORTD&=~(1<<led);  
    }  
}
```

e) Nenhuma das alternativas apresentadas.

Questão 10 (10 pontos) - No manual do ATMEGA328 pode-se verificar que o registrador “EIMSK” (*External Interrupt Mask Register*) é responsável por habilitar as interrupções nos pinos chamados de **INT0** e **INT1**. Sobre estes são feitas as seguintes afirmações:

I. Estas interrupções são ativadas somente quando as entradas a elas associadas assumem nível lógico 0 ou nível lógico 1.

II. São consideradas como “de alta prioridade” e estão associadas a todos os pinos de GPIO.

III. Possuem prioridade menor que a entrada de RESET do ATMEGA328.

Pode-se considerar como **verdadeira(s)** a(s) seguinte(s) afirmação(ões):

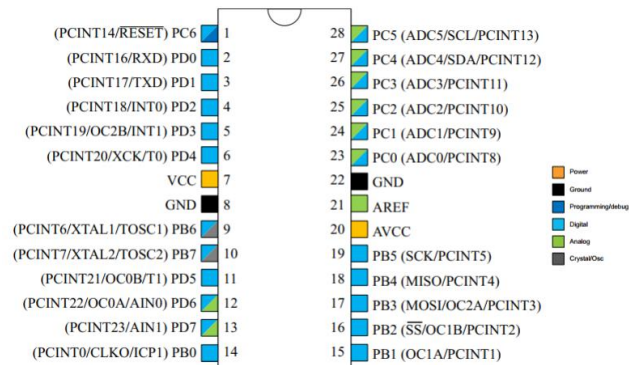
a) Apenas II.

b) Apenas I e II.

c) Apenas II e III.

d) Apenas III.

e) Nenhuma das alternativas.



Bit	7	6	5	4	3	2	1	0	
(0x69)	—	—	—	—	ISC11	ISC10	ISC01	ISC00	EICRA
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

INT1		Table 13-1. Interrupt 1 Sense Control		
		ISC11	ISC10	Description
		0	0	The low level of INT1 generates an interrupt request.
		0	1	Any logical change on INT1 generates an interrupt request.
		1	0	The falling edge of INT1 generates an interrupt request.
INT0		Table 13-2. Interrupt 0 Sense Control		
		ISC01	ISC00	Description
		0	0	The low level of INT0 generates an interrupt request.
		0	1	Any logical change on INT0 generates an interrupt request.
		1	0	The falling edge of INT0 generates an interrupt request.
		1	1	The rising edge of INT0 generates an interrupt request.

Bit	7	6	5	4	3	2	1	0	
0x1D (0x3D)	—	—	—	—	—	—	INT1	INT0	EIMSK
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
0x1C (0x3C)	—	—	—	—	—	—	INTF1	INTF0	EIFR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

RESPOSTAS:

NOME: _____

CURSO: _____ MATRÍCULA: _____

QUESTÃO	RESPOSTA:				
	A	B	C	D	E
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					