PROVA 2

MAC 2166 – INTRODUÇÃO À COMPUTAÇÃO (ELÉTRICA E COMPUTAÇÃO)

PRIMEIRO SEMESTRE DE 2026

Nome completo: _____

NUSP: _____ **Turma:** _____

Assinatura: _____

Instruções:

1. Não destaque as folhas deste caderno.
2. A prova pode ser feita a lápis. Cuidado com a legibilidade.
3. A prova consta de **8 páginas** (incluindo a atual), totalizando **3 questões**. Verifique, antes de começar a prova, se o seu caderno de questões está completo.
4. Preencha o cabeçalho acima e os campos NUSP e Turma das folhas seguintes.
5. Não é permitido o uso de folhas avulsas para rascunho.
6. Não é permitido o uso de artefatos eletrônicos de qualquer natureza.
7. Não é permitida a consulta a livros, apontamentos ou colegas.
8. Não é necessário apagar rascunhos no caderno de questões.

DURAÇÃO DA PROVA: 2 horas

Questão	Nota
1	
2	
3	
Total	

Q1. **[3 pontos]** Considere o seguinte código.

```
#include <stdio.h>

void f(char *c) {
    char ini;
    int s;
    if (*c >= 'A' && *c <= 'Z') {
        ini = 'A';
        s = 9;
    } else if (*c >= 'a' && *c <= 'z') {
        ini = 'a';
        s = 11;
    } else {
        return;
    }
    int n = *c - ini;
    n = (n + s) % 26;
    *c = n + ini;
}

int main(int argc, char *argv[]) {
    char ch;
    while (scanf("%c", &ch) != EOF) {
        f(&ch);
        printf("%c", ch);
    }
    return 0;
}
```

Lembre que a ordem das 26 letras do alfabeto é:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Determine a saída impressa por tal código para cada uma das seguintes entradas:

(a) `Sd@-Edxit!`

Saída:

(*b*) `Wdx+Ttgid?`

Saída:

(*c*) `Ktb/Jdgit#`

Saída:

NUSP: _____ **Turma:** _____

Q2. **[3 pontos]** Nesta questão você irá implementar um procedimento que recebe como entrada um vetor de inteiros, digamos `a`, e ordena-o (isto é, muda a ordem dos elementos para que tenhamos `a[i] <= a[j]` para quaisquer índices $i < j$). Não é necessário ter conhecimento prévio de algoritmos de ordenação.

(a) Escreva uma função `int indiceMaximo(int a[], int n)` que considera os `n` primeiros elementos do vetor `a` e devolve o índice do elemento de valor máximo. Caso haja empate, sua função deve devolver o índice da última ocorrência do valor máximo. Lembre-se que índices começam de 0.

(b) Escreva uma função `void ordenaSelecao(int a[], int n)` que use a função do item anterior repetidamente para ordenar o vetor `a`, que tem `n` elementos. Sua função inicialmente colocará (através de uma troca) o maior elemento na última posição do vetor `a`, onde ele ficará durante o resto da execução. Em seguida, colocará o segundo maior elemento na penúltima posição, e assim por diante. Ao final da execução da sua função, o vetor `a` deverá estar ordenado.

Q3. **[4 pontos]** Numa empresa, funcionários batem ponto ao começar o dia de trabalho e ao terminá-lo. Sua tarefa nesta questão é implementar um relatório do sistema de ponto que mostre quantas horas cada pessoa trabalhou num dado dia.

(*a*) Implemente uma função `int numSegundos(int hh, int mm, int ss)` que recebe três inteiros que correspondem a um horário no formato hh:mm:ss. Ela deve devolver o número de segundos transcorridos entre a meia-noite (00:00:00) e o horário dado. Por exemplo, `numSegundos(0, 1, 13)` deve devolver 73, pois o intervalo de tempo entre meia-noite e 00:01:13 tem duração de 73 segundos.

(*b*) Implemente uma função

```
void horaMinSeg(int num_seg, int *hh, int *mm, int *ss)
```

que recebe um inteiro $0 \leq$ `num_seg` < 86400 e calcula números `hh`, `mm` e `ss` tais que hh:mm:ss é o horário obtido contando-se `num_seg` segundos a partir da meia-noite (isto é, esta função é a "inversa" da função do item anterior).

NUSP: _____ **Turma:** _____

(*c*) Implemente a função `main` do seu programa. Ela deve ler linhas até o fim do arquivo (EOF). As linhas terão o formato `hh mm ss identificador`, em que $0 \leq$ `hh` < 24, $0 \leq$ `mm` < 60, $0 \leq$ `ss` < 60 e $0 \leq$ `identificador` < 100. Abaixo temos um exemplo de entrada, que explicaremos em seguida.

```
8 10 9 3
8 30 2 4
9 0 40 1
16 0 20 1
17 0 10 3
17 20 1 4
```

Por exemplo, a linha `8 10 9 3` quer dizer que, às 8:10:09, a pessoa de identificador 3 bateu ponto. Cada pessoa é identificada por um número diferente no intervalo $[0, 100)$.

É garantido que as linhas são dadas em ordem crescente de horário. Toda pessoa da empresa ocorre em exatamente duas linhas: a primeira para registrar o começo do dia de trabalho da pessoa, e a segunda para registrar o fim do dia. Note que nem todo identificador no intervalo $[0, 100)$ ocorre na entrada (algumas pessoas podem não trabalhar no dia).

Seu programa deve imprimir uma linha por pessoa, dizendo quanto ela trabalhou, no formato exemplificado abaixo. As linhas devem ser impressas em ordem crescente de identificador. Para a entrada dada nas 6 linhas do exemplo acima, seu programa deve imprimir a saída exatamente como abaixo:

```
Pessoa 1 trabalhou 6 hora(s), 59 minuto(s) e 40 segundo(s).
Pessoa 3 trabalhou 8 hora(s), 50 minuto(s) e 1 segundo(s).
Pessoa 4 trabalhou 8 hora(s), 49 minuto(s) e 59 segundo(s).
```

Seu programa deve usar as funções dos itens anteriores (mesmo que você não os tenha resolvido). Note que não é necessário distinguir singular de plural na sua saída.

Espaço extra para a solução da Questão 3:

NUSP: _____ **Turma:** _____

(Rascunho)