

INF 1007 – P1A – 26/09/09	Questão 1
Nome:	Estação:
Matrícula:	Turma

Em um estacionamento rotativo, que funciona diariamente das 6 às 22 horas, a tarifa a ser paga por cada cliente é calculada em função do número de horas que o veículo permanece no local, segundo a tabela de custo por hora abaixo.

Tempo	Custo (R\$)
1ª hora	8,00
2ª hora	6,00
3ª hora	3,50
4ª e 5ª horas	3,00 por hora
A partir da 6ª hora	2,00 por hora

O tempo de permanência é calculado como um valor inteiro de horas, arredondando-se para cima qualquer fração. Por exemplo, um veículo que entra às 7:35h e sai às 18:10h tem tempo de permanência de 11h. A tarifa corresponde a soma de R\$8, pela primeira hora, R\$6, pela segunda hora, R\$3,50, pela terceira hora, 2 x R\$3, pelas quarta e quinta horas, e 6 x R\$2, pelo tempo restante, num total de R\$35,50.

a) [Valor: 4,0 pontos] Escreva uma função em C que recebe 5 parâmetros: os inteiros $h1$, $m1$, $h2$ e $m2$, representando, respectivamente, a hora e minuto da entrada e a hora e minuto da saída de um veículo, e o ponteiro para inteiro $tempo$. Considere que a saída é sempre posterior à entrada. A função deve calcular o tempo de permanência, que deve ser armazenado no endereço indicado pelo parâmetro $tempo$, e calcular e retornar o valor da tarifa devida, de acordo com o protótipo abaixo.

```
float calc_tarifa(int h1, int m1, int h2, int m2, int *tempo);
```

b) [Valor: 2,0 pontos] Escreva um programa completo em C que:

1. lê do teclado quatro números inteiros, representando, respectivamente, a hora e minuto da entrada e a hora e minuto da saída de um veículo;
2. em seguida, utilizando a função desenvolvida no item “a”, determina o tempo de permanência e a tarifa a ser paga e imprime esses valores na tela.

Veja a seguir um exemplo da saída esperada:

```
7 35 18 10
11 horas
32.50 reais
```

INF 1007 – P1A – 26/09/09	Questão 2
Nome:	Estação:
Matrícula:	Turma

[Valor: 4,0 pontos] Para evitar que estranhos tenham acesso ao conteúdo de seus arquivos de texto, um programador criou um esquema especial para armazená-los. Nesse esquema, cada palavra é subdividida em duas cadeias de caracteres que são armazenadas separadamente. A primeira cadeia é formada com os caracteres de ordem ímpar (primeiro, terceiro, quinto, etc) e a outra com os caracteres de ordem par (segundo, quarto, sexto, etc) da palavra original. Por exemplo, a palavra “PARALELEPIEDO” seria subdividida nas cadeias “PRLPPD” e “AAEEIEO”. Posteriormente, para obter cada palavra original, basta mesclar as duas cadeias correspondentes tomando os caracteres na ordem correta.

Escreva uma função que receba como parâmetros os ponteiros para as duas cadeias de caracteres $s1$ e $s2$, contendo respectivamente os caracteres de ordem ímpar e de ordem par da palavra original, respectivamente, e retorne um ponteiro para uma nova cadeia de caracteres, alocada dinamicamente, que contenha a palavra completa. Se não for possível criar a nova cadeia, a função deve retornar NULL. Considere que as cadeias fornecidas como entrada contêm apenas caracteres correspondentes a letras maiúsculas de 'A' a 'Z'. O protótipo da função é:

```
char* mesclar(char* s1, char* s2);
```

RASCUNHO

Respostas nesta folha não serão consideradas.

Protótipos de funções que podem ser úteis:

stdio.h:

```
int scanf (char* formato, ...);
int printf (char* formato, ...);
FILE* fopen (char* nome, char* modo);
int fclose (FILE* fp);
int fscanf (FILE* fp, char* formato, ...);
int fprintf (FILE* fp, char* formato, ...);
char* fgets(char* str, int size, FILE* fp);
int sscanf(char* str, char* formato, ...);
```

math.h:

```
double sqrt (double x);
double pow (double x, double exp);
double cos (double radianos);
double sin (double radianos);
```

string.h:

```
int strlen (char* s);
int strcmp (char* s, char *t);
char* strcpy (char* destino, char* origem);
char* strcat (char* destino, char* origem);
```

stdlib.h:

```
void* malloc (int nbytes);
void free (void* p);
void qsort (void *vet, int n, int tam, int (*comp) (const void*, const void*));
void* bsearch (void* info, void *v, int n, int tam, int (*cmp)(const void*,
                                                             const void*));
```

Não separe as folhas deste caderno.