Primeira Prova de Linguagens de Programação
- DCC024B -
Sistemas de Informação

Nome: _____
"Eu dou minha palavra de honra que não trapacearei neste exame."

Número de matrícula: _____

As regras do jogo:

- A prova é sem consulta.
- Quando terminar, não entregue nada além do caderno de provas para o instrutor.
- Quando escrever código, a sintaxe correta é importante.
- Cada estudante tem direito a fazer uma pergunta ao instrutor durante a prova. Traga o caderno de provas quando vier à mesa do instrutor.
- A prova termina uma hora e quarenta minutos após seu início.
- Seja honesto e lembre-se: **você deu sua palavra de honra**.

Alguns conselhos:

- Escreva sempre algo nas questões, a fim de ganhar algum crédito parcial.
- Se não entender a questão, e já tiver gasto sua pergunta, escreva a sua interpretação da questão junto à resposta.
- A prova não é difícil, ela é divertida, então aproveite!

Tabela 1: Pontos acumulados (para uso do instrutor)

Questão 1	Questão 2	Questão 3	Extra

Questão Extra (0.5 Pontos): como construir quatro triângulos equiláteros com seis palitos de fósforo indeformáveis? Resposta em uma palavra:

1. A "cardinalidade" de um tipo é o número de instâncias daquele tipo. Em cada questão abaixo, informe a cardinalidade do tipo `T`. Você pode usar a cardinalidade dos tipos constituintes em sua resposta (a cardinalidade do tipo `int`, do tipo `bool`, etc).

   (a) (1.25 Pontos) Em SML/NJ: `datatype T = I of int | R of real`

   (b) (1.25 Pontos) Em SML/NJ: `type T = int * real`

   (c) (1.25 Pontos) Em SML/NJ: `type T = bool list`

   (d) (1.25 Pontos) Em SML/NJ: `datatype 'a T = NONE | SOME of 'a`

   (e) (1.25 Pontos) Em SML/NJ: `datatype T = Sat | Sun`

   (f) (1.25 Pontos) Em C: `typedef struct { int i; char c; } T;`

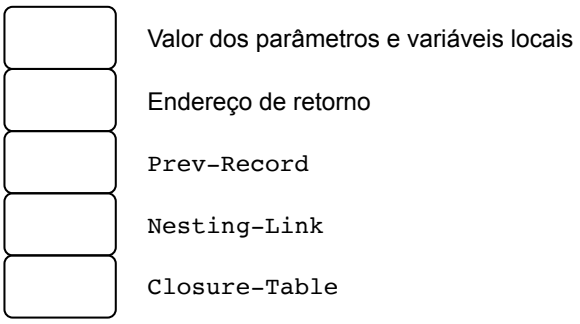
   (g) (1.25 Pontos) Em C: `typedef union { int i; char c; } T;`

   (h) (1.25 Pontos) Em C: `enum T { LOW, MEDIUM, HIGH };`

2. (0.5 Pontos cada) Registros de ativação são regiões de memória que guardam as informações necessárias à ativação de funções. Registros de ativação incluem diferentes tipos de dados, dependendo de como a linguagem é implementada. Exemplos de dados armazenados em registros de ativação incluem: endereço de retorno da função, valor dos parâmetros, valor de retorno, valor das variáveis locais, ponteiro para o registro de ativação da função anteriormente ativa (`Prev-Record`), ponteiro para o registro de ativação da função aninhadora (`Nesting-Link`), ponteiro para a tabela de variáveis livres na função (`Closure-Table`). Em cada figura abaixo, diga quais dessas informações devem estar presentes no registro de ativação de cada linguagem de programação.

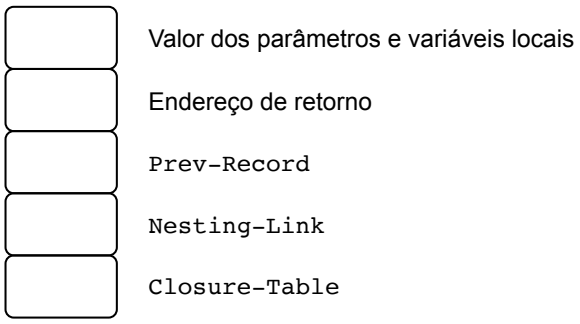
A linguagem Fortran 66, que somente possuia alocação estática de memória

```
FUNCTION ADDITION(X, Y)
REAL X, Y, ADDITION
ADDITION = X + Y
RETURN
END
```



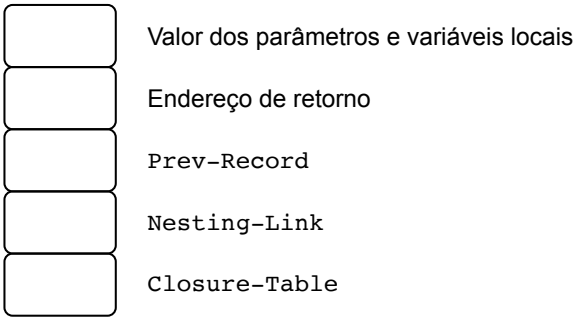
A linguagem ANSI C padrão, que não permite funções aninhadas.

```
int main(int argc, char** argv) {
  int x = argc - 1;
  printf("Number of args = %d\n", x);
  return 0;
}
```



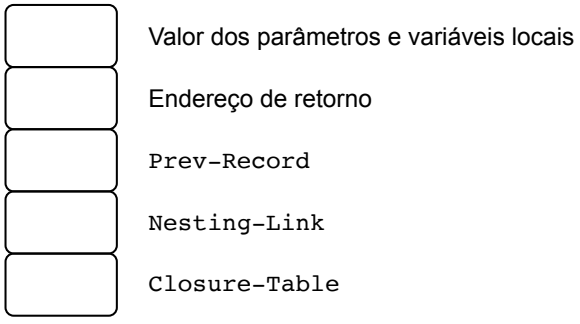
A linguagem C compilada pelo compilador gcc, que suporta funções aninhadas:

```
int outerFunction(int a) {
  int innerFunction(int b) {
    return b * 2;
  }
  return innerFunction(a);
}
```



A linguagem SML/NJ que vimos em sala de aula, que permite retornar funções aninhadas.

```
fun funToAddX x =
  let
    fun addX y = y + x
  in
    addX
  end
```



Em cada caso acima, escreva no retângulo correspondente a uma informação a letra S, caso a informação esteja presente no registro de ativação, ou a letra N, caso aquela informação não esteja presente.

3. O cálculo da mediana é uma maneira de encontrar um valor central em um conjunto de dados. Existe um algoritmo linear para encontrar a mediana de uma lista de números: (1) Escolha um elemento *pivot*. (2) Particione a lista em dois subconjuntos: um contendo elementos menores que o *pivot* e outro contendo elementos maiores. (3) Se o índice do *pivot* for a mediana, retorne o valor do *pivot*. (4) Caso contrário, recorra no subconjunto apropriado com base na relação entre o índice da mediana e o índice do *pivot*. Abaixo temos uma implementação deste algoritmo em Python. Nesta questão, você deve traduzir o algoritmo em Python para um conjunto de três funções equivalentes (`split`, `select` e `median`) em SML/NJ.

Código escrito em Python que encontra a mediana de uma lista de N elementos em O(N):

3 Pontos

```
def split(lst, pivot):
    small = [x for x in lst if x < pivot]
    large = [x for x in lst if x > pivot]
    return small, large
```

3 Pontos

```
def select(k, lst):
    if len(lst) == 1:
        return lst[0]
    pivot = lst[0]
    small, large = split(lst[1:], pivot)
    s_len = len(small)
    if k == s_len:
        return pivot
    elif k < s_len:
        return select(k, small)
    else:
        return select(k - s_len - 1, large)
```

4 Pontos

```
def median(lst):
    n = len(lst)
    if n % 2 == 1:
        return select(n // 2, lst)
    else:
        left = select(n // 2 - 1, lst)
        right = select(n // 2, lst)
        return (left + right) / 2
```

```
# Example (não precisa traduzir esta parte):
>>> lst = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5]
>>> median(lst) == 5
True
```

Cheat Sheet

Python:	**SML/NJ**:
`len(L)`	`length L`
`a // b`	`a div b`
`a % b`	`a mod b`

Código equivalente em SML/NJ