1.Uma introdução a biblioteca <stdlib.h>

Este cabeçalho define diversas funções de uso geral, incluindo gerenciamento dinâmico de memória, geração de números aleatórios, comunicação com o ambiente, aritmética de inteiros, pesquisa, classificação e conversão.

A função ***malloc*** aloca um bloco de memória de tamanho variável e retorna um ponteiro para o início deste bloco. O conteúdo deste novo bloco de memória não está inicializado, permanecendo com valores indeterminados. Se o valor do bloco alocado for 0, o valor de retorno depende da implementação da biblioteca padrão ANSI. O ponteiro não é desreferenciado.

```
int main (){

int *valuePtr;

valuePtr = (void*) malloc (tamanhoBloco);

return (0);

}
```

A função ***rand*** gera um inteiro entre 0 e RAND_MAX, isto é, uma constante simbólica definida no arquivo de cabeçalho ***stdlib.h***. O padrão ANSI estabelece que o valor RAND_MAX deve ser menor que 32.767, que é o valor máximo de um inteiro de 2 bytes. Se ***rand*** produzir valores inteiros verdadeiramente aleatórios, todos os números entre 0 e RAND_MAX terão a mesma probabilidade de serem escolhidos cada vez que a função for chamada. Na realidade, esta função gera números pseudo-aleatórios.

Chamar a função repetidamente produz uma sequência de números que parece ser aleatória, entretanto, a sequência se repete cada vez que o programa é executado. Depois do programa ser completamente depurado, ele pode ser condicionado de modo a produzir uma sequência diferente de números em cada execução. Isso é chamado de randomização, e é realizado com a função ***srand()*** da biblioteca padrão. A função utiliza um argumento ***unsigned int*** para ser a *seed* da função ***rand()*** de forma que seja produzida uma sequência diferente de números aleatórios em cada execução do programa.

Executando o programa várias vezes, uma sequência diferente de números aleatórios é obtida cada vez que o programa é executado já que, em cada uma delas, uma semente diferente é fornecida. Se desejássemos randomizar sem a necessidade de fornecer uma semente a cada vez, poderíamos usar uma instrução como ***srand(time(NULL))***. Isso faz com que o computador leia seu relógio para obter automaticamente o valor da semente.

A função ***time()*** retorna a hora atual do dia em segundos. Esse valor é convertido em um ***unsigned int*** que é usado como *seed* do gerador de números aleatórios.