Atividade diária dia 25 de julho de 2018

**01)O que é a biblioteca padrão do C++? Pesquise 20 funções básicas.**

R:  A biblioteca padrão é uma coleção de classes, funções e variáveis escritas na própria linguagem para facilitar o desenvolvimento de aplicações.

20 funções básicas do C++ são

1. #include time.h  
   Esta biblioteca declara funções para manipularmos a data e hora
2. <cassert> (assert.h)

Biblioteca de diagnósticos C

1. <cctype> (ctype.h)

Funções de manipulação de caracteres

1. assert.h

Macro para diagnóstico.

1. string.h

rotinas p/ manipular strings e memória

1. stdlib.h

várias rotinas muito usadas, conversão, sort, etc.

1. stddef.h

vários tipos de dados e macro substituições

1. math.h

funções matemáticas

1. float.h

funções para tratar números de ponto flutuante

1. alloc.h

funções para gerenciamento de memória

1. stdio.h

rotinas padrão de entrada e saída definidas pelos criadores da linguagem C.

1. locale.h

Biblioteca de localização C

1. <vector>

Cabeçalho de veto

1. cwctype> (wctype.h)

Tipo de personagem amplo

1. <cuchar> (uchar.h)

Caracteres Unicode

1. <ctime> (time.h)

Biblioteca C Time

1. <cstring> (string.h)

C Cordas

1. <cstdint> (stdint.h)

Tipos inteiros

1. <cstdarg> (stdarg.h)

Manipulação de argumentos variáveis

1. <cmath> (math.h)

Biblioteca numérica C

**02) Qual a versão mais recente do C++? Quais as principais mudanças em relação a versão anterior?**

R: a versão mais atual do C++ é a versão C++17, que é uma versão melhorada do C++14, esta que é uma versão melhorada do C++11 em relação ao C++14 a sua versão anterior não foram adicionada muitas mudanças.

C ++ 14 adiciona ao C ++ 11:

* Restrições constexpr relaxadas (o recurso mais caro do C ++ 14 para implementar, no meu caso)
* Lambdas genéricos (por exemplo )[](auto p) { return p\*2; }
* Captura de inicialização (por exemplo, )[i = 2](auto p) { return p+i++; }
* Modelos variáveis
* decltype(auto)
* Tipos de retorno deduzidos
* Literais binários (por exemplo, 0b11101100)
* Separadores de dígitos (por exemplo, )0b1110'1100
* [[deprecated]] e [[deprecated("msg")]]
* Estender “tipo de classe agregada” para incluir uma classe que seria um tipo agregado de C ++ 11, se os inicializadores de membro padrão fossem omitidos

O C ++ 17 acrescenta isso:

* if constexpr (...)
* Declarações de ligação estruturada (por exemplo, )auto [x, y] = f();
* Elaboração de cópia obrigatória através de reformulação de categorias de valor
* Variáveis ​​embutidas (permitindo múltiplas definições nas unidades de tradução)
* Ordem mais forte de garantias de avaliação
* Dobre expressões (por exemplo, )(pack + … + 0)
* Definições de namespace aninhadas (por exemplo, )namespace Outer::Inner {}
* Trigraphs se foram (yay!)
* Init-demonstrações no if/ / (por exemplo, )if constexprswitchif (auto x = f(); !x.empty()) …
* Dedução de argumento de modelo de classe (por exemplo, )pair p = { true, 1.0 };
* Literais hexadecimais de ponto flutuante (por exemplo, )0x3.ABCp-10
* u8literais de caracteres (por exemplo, )u8'x'
* Generalização de argumentos de modelo nontype (por exemplo, onde denota um membro de dados estático pode agora ser válido)X<&s.s>s.s
* Funções de alocação / desalocação super alinhadas ( etc.)operator new(size\_t, align\_val\_t)
* Atributos em namespaces
* Atributos nas constantes do enumerador
* [[nodiscard]], e[[maybe\_unused]][[fallthrough]]
* typenameé permitido para declarações de parâmetro de modelo de modelo (por exemplo, )template<template<typename> typename X> struct …
* static\_assert não requer mais um segundo operando (string literal).
* Ajustar às regras para deduzir autocom um inicializador armado
* \_\_has\_include

**03) Quais as melhorias de desempenho de tempo de execução da linguagem padrão, temos nessa nova versão do C++?**

**R:** os novos recursos foram adicionado com o intuito de melhorar o desempenho tornado os códigos mais vividos em comparação com sua versão anterios essa mudanças foram, loops baseados em intervalos, listas de inicializadores, lambdas e ponteiros inteligentes.