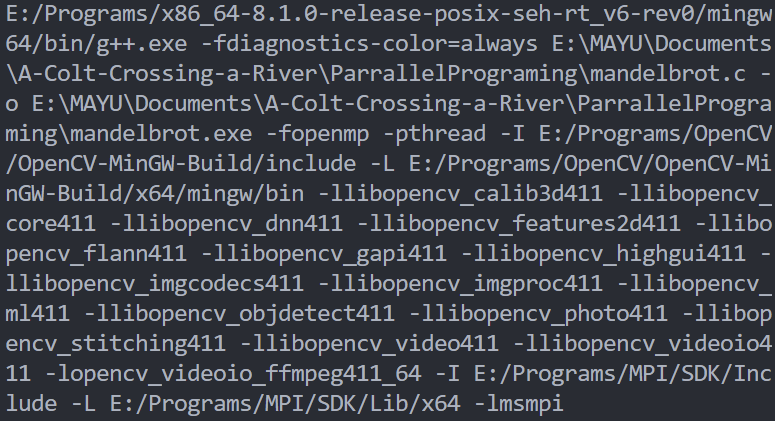
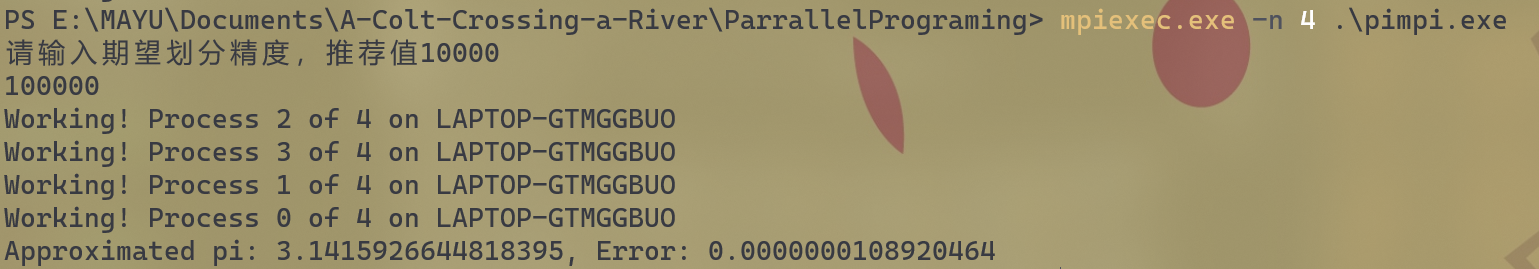
编译命令如下，因为用的vscode，之前编译用的链接库懒得删了



运行和结果



代码

#include <stdio.h>

#include <iostream>

#include <mpi.h>

#include <math.h>

#ifndef **M\_PI**

#define **M\_PI** (3.14159265358979323846)

#endif

int **main**(int *argc*, char \**argv*[]) {

    int rank, size;

**MPI\_Init**(&*argc*, &*argv*);

**MPI\_Comm\_rank**(**MPI\_COMM\_WORLD**, &rank);

**MPI\_Comm\_size**(**MPI\_COMM\_WORLD**, &size);

    int n;

    double mypi = 0.0;

    if (rank == 0) {

        std::cout**<<**"请输入期望划分精度，推荐值10000"**<<**std::**endl**;

        std::cin**>>**n;

    }

**MPI\_Bcast**(&n, 1, **MPI\_INT**, 0, **MPI\_COMM\_WORLD**);

    double w = 1.0 / (double)n;

    double sum = 0.0;

    for (int i = rank + 1; i <= n; i += size) {

        double x = w \* ((double)i - 0.5);

        sum += **sqrt**(1.0 - x \* x);

    }

    double pi;

    int my\_rank, numprocs;

**MPI\_Comm\_size**(**MPI\_COMM\_WORLD**,&numprocs);

**MPI\_Comm\_rank**(**MPI\_COMM\_WORLD**,&my\_rank);

    int  namelen;

    char processor\_name[**MPI\_MAX\_PROCESSOR\_NAME**];

**MPI\_Get\_processor\_name**(processor\_name,&namelen);

**printf**("Working! Process %d of %d on %s\n",my\_rank,numprocs,processor\_name);

**MPI\_Reduce**(&sum, &pi, 1, **MPI\_DOUBLE**, **MPI\_SUM**, 0, **MPI\_COMM\_WORLD**);

    if (rank == 0) {

        pi \*= 4.0 \* w;

**printf**("Approximated pi: %.16f, Error: %.16f\n", pi, **fabs**(pi - **M\_PI**));

    }

**MPI\_Finalize**();

    return 0;

}