Computing & Programming — Year 2025–2026, Practice 2

Compiler gcc and archiver ar — (Date: –/–/2025)

1.– Create a folder named C on the hard drive.
Navigate to the folder and create the text files `prog.c`, `sub1.c`, `sub2.c` and `sub3.c`.
Please refer to the C program listed on the following pages.
Using a Wordpad-type application or an editor such as Geany, SciTe, Emacs, or Vim, write the main() function in the file. `prog.c` and the Generate(), Alter(), and Print_sol() functions will be written in the files `sub1.c`, `sub2.c` and `sub3.c` respectively.

You are asked to:

a) Examine, analyze, and explain the operation of the main program and each of the subroutines.

b) Build the executable program `prog.exe` using the instructions

```
$ gcc prog.c sub1.c sub2.c sub3.c -O2 -o prog.exe
```

and verify that it works correctly.

c) Delete the executable program `prog.exe` and rebuild it using the instructions

```
$ gcc sub1.c -O2 -c -o sub1.o
$ gcc sub2.c -O2 -c -o sub2.o
$ gcc sub3.c -O2 -c -o sub3.o

$ gcc prog.c -O2 -c -o prog.o
$ gcc prog.o sub1.o sub2.o sub3.o -O2 -o prog.exe

$ del prog.o sub1.o sub2.o sub3.o
```

and verify that it works correctly.

d) Delete the executable program `prog.exe` and rebuild it using the instructions

```
$ gcc sub1.c -O2 -c -o sub1.o
$ ar r libsubs.a sub1.o
$ del sub1.o

$ gcc sub2.c -O2 -c -o sub2.o
$ ar r libsubs.a sub2.o
$ del sub2.o

$ gcc sub3.c -O2 -c -o sub3.o
$ ar r libsubs.a sub3.o
$ del sub3.o

$ gcc prog.c -O2 -c -o prog.o
$ gcc prog.o libsubs.a -O2 -o prog.exe

$ del prog.o
```

and verify that it works correctly.

```c
/* Primitive style program */

#include <stdio.h>
#include <stdlib.h>
#define MX 1000

int main(void)
{
     void Generate(int, double [ ]);
     void Alter(int, double [ ]);
     void Print_sol(int, double [ ]);

     int n;
     double v[MX];

     ReadNComponents:
          printf("\n Indicate the number of components: ");
          scanf("%d", &n);
     if (n < 1 || n > MX) goto ReadNComponents;

     Generate(n, v);
     Alter(n, v);
     Print_sol(n, v);

     fflush(stdin); getchar( );
     exit(0);
}
```

```
/*_____________________________________________________________
Generate function, difficult to understand and whose purpose is unknown

It uses the functions of the standard library:

     void srand(unsigned int iseed)
                    -> it uses iseed as seed of a sequence
                         of pseudo-random numbers
     int rand(void)
                    -> it returns a pseudo-random number
                         (integer) in the range 0-RAND_MAX
*/

#include <stdio.h>
#include <stdlib.h> /* It declares the functions srand( ), rand( ) */

void Generate(int n, double v[ ])
{
     int iseed,
          i;

     ReadSeed:
          printf("\n Indicate the seed (integer number > 0): ");
          scanf("%d", &iseed);
     if (iseed < 0 ) goto ReadSeed;

     srand(iseed);
     i = 0;
     Generate:
          v[i] = (double) ((float) ((double) rand( ) / (double)
RAND_MAX));
          i++;
     if (i < n) goto Generate;
}
```

```
/*_____________________________________________________________
Alter function, difficult to understand and whose purpose is unknown
*/

void Alter(int n, double v[ ])
{
     int i, j;
     double vc;

     i = 0;
     Externalloop:
          j = n - 1;
          Internalloop:
               if (v[i] <= v [j]) goto Continue;
                    vc = v[i];
                    v[i] = v[j];
                    v[j] = vc;
               Continue:
               j--;
          if (j > i) goto Internalloop;
          i++;
     if (i < n - 1) goto Externalloop;
}
```

```
/*_____________________________________________________________
Print function, difficult to understand and whose purpose is unknown
*/

#include <stdio.h>
#include <stdlib.h>
void Print_sol(int n, double v[ ])
{
      int i;

      printf("\n\n RESULT:\n\n");

      i = 0;
      Writing:
            printf(" v[%5d] = %20.9f\n", i, v[i]);
            i++;
      if (i < n) goto Writing;
}
```