

```

1 //FOLDER: sequential_example_explicit
2
3 /*
4  * This is an example program for PGAPack. The objective is to maximize the
5  * function  $y=x^2$  in  $[0,255]$ .
6  */
7
8 #include <pgapack.h>
9
10 #define INDLEN 16
11
12 double EvaluationFunction(PGAContext *, int, int);
13
14 /*****
15  *          main program
16  *****/
17 int main( int argc, char **argv ) {
18     PGAContext *ctx;
19
20     ctx = PGACreate(&argc, argv, PGA_DATATYPE_BINARY, INDLEN, PGA_MAXIMIZE);
21     PGASetPopSize(ctx, 20);
22     PGASetMaxGAIterValue(ctx, 100);
23     PGASetPrintFrequencyValue(ctx, 1);
24     PGASetRandomSeed(ctx, 1);
25     PGASetUp(ctx);
26
27     //PGARun(ctx, EvaluationFunction)
28     PGAEvaluate(ctx, PGA_OLDPOP, EvaluationFunction, 0);
29     PGAFitness(ctx, PGA_OLDPOP);
30     while(!PGADone(ctx, 0))
31     {
32         PGASelect                (ctx, PGA_OLDPOP);
33         PGARunMutationAndCrossover (ctx, PGA_OLDPOP, PGA_NEWPOP);
34         PGAEvaluate                (ctx, PGA_NEWPOP, EvaluationFunction, 0);
35         PGAFitness                (ctx, PGA_NEWPOP);
36         PGAUpdateGeneration        (ctx, 0);
37         PGAPrintReport             (ctx, stdout, PGA_OLDPOP);
38     }
39     PGADestroy(ctx);
40
41     return(0);
42 }
43
44 /*****
45  * user defined evaluation function
46  * ctx - contex variable
47  * p - chromosome index in population
48  * pop - which population to refer to
49  *****/
50 double EvaluationFunction(PGAContext *ctx, int p, int pop) {
51     int int_val, stringlen;
52
53     stringlen = PGAGetStringLength(ctx);
54     int_val = PGAGetIntegerFromBinary(ctx, p, pop, 0, stringlen-1);
55
56     return((double) int_val*int_val);
57 }

```