

```
1 //FOLDER: sequential_example_holland
2
3 /*
4  * This is an example program for PGAPack. The objective is to maximize the
5  * function  $y=x^2$  in  $[0,2^{16}-1]$ .
6  */
7
8 #include <pgapack.h>
9
10 #define INDLEN 16
11 #define POPSIZE 20
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, POPSIZE);
22
23     PGASetSelectType(ctx, PGA_SELECT_PROPORTIONAL);
24     PGASetNumReplaceValue(ctx, POPSIZE);
25     PGASetCrossoverType(ctx, PGA_CROSSOVER_ONEPT);
26     PGASetCrossoverProb(ctx, 1.0);
27     PGASetMutationType(ctx, PGA_MUTATION_CONSTANT);
28     PGASetMutationProb(ctx, 1/INDLEN);
29
30     PGASetMaxGAIterValue(ctx, 100);
31     PGASetPrintFrequencyValue(ctx, 1);
32     PGASetRandomSeed(ctx, 1);
33
34     PGASetUp(ctx);
35     PGARun(ctx, EvaluationFunction);
36     PGADestroy(ctx);
37
38     return(0);
39 }
40
41 /*****
42  * user defined evaluation function
43  * ctx - contex variable
44  * p - chromosome index in population
45  * pop - which population to refer to
46  *****/
47 double EvaluationFunction(PGAContext *ctx, int p, int pop) {
48     int int_val, stringlen;
49
50     stringlen = PGAGetStringLength(ctx);
51     int_val = PGAGetIntegerFromBinary(ctx, p, pop, 0, stringlen-1);
52
53     return((double) int_val*int_val);
54 }
```