**ГЛАВА 16**

**1.**

а.

#define FPM 5280

d i s t = FPM \* m i l e s;

Код допустим, результатом будет умножение двух чисел, FPM заменится числом 5280.

б.

#define FEET 4  
#define POD FEET + FEET

plort = FEET \* POD;

Код недопустим, так как, первой будет операция умножения и в результате получится 4 \* 4 + 4 = 20, а должно быть 4 \* (4 + 4) = 32. Для устранения такой ошибки, нужно макрос POD заключить в круглые скобки #define POD (FEET + FEET), ведь препроцессор только делает подстановку выражений в код, но не их расчет!

в.

#define SIX = 6;

nex = SIX;

Код недопустим, компилятор выдаст ошибку, потому, что нельзя к макросу применять операцию присваивания и не нужно в конце строки ставить точку с запятой, ведь определение макроса обрабатывается до символа новой строки.

г.

#define NEW(X) X + 5

y = NEW(y); - допустимо, результат будет y = y + 5

berg = NEW(berg) \* lob; - недопустимо, результат berg = berg + 5 \* lob, нужно использовать круглые скобки в определении функционального макроса, чтобы сначала обрабатывалась операция сложения, а после уже умножения, результат был berg = (berg + 5) \* lob

est = NEW(berg) / NEW(y); - недопустимо, результат est = berg + 5 / y + 5, также нужно использовать круглые скобки в определении макроса est = (berg + 5) / (y + 5)

nilp = lob \* NEW(-berg); - недопустимо, результат nilp = lob \* -x + 5, нужно использовать круглые скобки в определении макроса, nilp = lob \* (-berg + 5).

**2.**

#define NEW(X) ((X) + 5)

**3.**

#define LOWER(A, B) ((A) < (B) ? (A) : (B))

**4.**

#define EVEN\_GT(X, Y) ((((X) % 2) == 0) && ((X) > (Y)) ? 1 : 0)

**5.**

#define SAMPLE(A, O, B) printf(“%d %c %d = %d\n”, A, O, B, A + B); не правильно

**6.**

a. #define NAMED 25

б. #define SPACE ‘ ‘

в. #define PS() printf(“%c”, ‘ ‘) – не правильно, нужно putchar(‘ ‘)

г. #define BIG(X) ((X) + 3)

д. #define SUMQ(X, Y) ((X) \* (X) + (Y) \* (Y))

**7.**

#define PRINT\_INT(X) printf(“имя: “ #X ”; значение: %d; адрес: %p\n”, X, &X)

**8.**

Закомментировать этот блок кода.

**12.**

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

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

{

double num = atof(argv[1]);

printf(“Квадратный корень из %f равен %f\n”, num, sqrt(num));

return 0;

}

**13.**

a. qsort(scores, 1000, sizeof(int), comp);

б. int comp(const void \* val\_1, const void \* val\_2)

{

if (\*val\_1 > \*val\_2)

return 1;

else if (\*val\_1 < \*val\_2)

return -1;

else

return 0;

}

**14.**

a. memcpy(data1, data2, sizeof(double) \* 100);

б. memcpy(data1, data2[199], sizeof(double) \* 100);