#include "main.h"

/\*\*

\* \_isupper - checks for uppercase character

\* @c: Variable text

\* Return: Always 0.

\*/

int \_isupper(int c)

{

if (c >= 'A' && c <= 'Z')

{

return (1);

}

else

{

return (0);

}

}

1

#include "main.h"

/\*\*

\* \_isdigit - checks for a digit (0 through 9)

\* @c: int to be checked

\* Return: 1 if c is a digit, 0 otherwise

\*/

int \_isdigit(int c)

{

return (c >= '0' && c <= '9');

}

2

#include "main.h"

/\*\*

\* mul - checks for checks for a digit (0 through 9).

\* @a: a - Variable

\* @b: b - variable

\* Return: Always 0.

\*/

int mul(int a, int b)

{

int mul;

mul = a \* b;

return (mul);

}

#include "main.h"

/\*\*

\* print\_numbers - checks for checks for a digit (0 through 9).

\*

\* Return: Always 0.

\*/

void print\_numbers(void)

{

int c;

for (c = 48; c < 58; c++)

{

\_putchar(c);

}

\_putchar('\n');

}

#include "main.h"

/\*\*

\* print\_numbers - checks for checks for a digit (0 through 9).

\*

\* Return: Always 0.

\*/

void print\_numbers(void)

{

int c;

for (c = 48; c < 58; c++)

{

\_putchar(c);

}

\_putchar('\n');

}

3

4

#include "main.h"

/\*\*

\* print\_most\_numbers - checks for checks for a digit (0 through 9).

\*

\* Return: Always 0.

\*/

void print\_most\_numbers(void)

{

int c;

for (c = 48; c < 58; c++)

{

if (c != 50)

{

if (c != 52)

{

\_putchar(c);

}

}

}

\_putchar('\n');

}

#include "main.h"

/\*\*

\* more\_numbers - prints 10 times the numbers, from 0 to 14

\* followed by a new line

\*/

void more\_numbers(void)

{

int i, j;

for (i = 0; i < 10; i++)

{

for (j = 0; j < 15; j++)

{

if (j >= 10)

\_putchar(j / 10 + '0');

\_putchar(j % 10 + '0');

}

\_putchar('\n');

}

}

6

#include "main.h"

/\*\*

\* print\_line - draws a straight line in the terminal

\* @n: number of times the character \_ should be printed

\*/

void print\_line(int n)

{

if (n <= 0)

{

\_putchar('\n');

} else

{

int i;

for (i = 1; i <= n; i++)

{

\_putchar('\_');

}

\_putchar('\n');

}

}

#include "main.h"

/\*\*

\* print\_diagonal - draws a diagonal line on the terminal

\* @n: number of times the character \ should be printed

\*/

void print\_diagonal(int n)

{

if (n <= 0)

{

\_putchar('\n');

} else

{

int i, j;

for (i = 0; i < n; i++)

{

for (j = 0; j < n; j++)

{

if (j == i)

\_putchar('\\');

else if (j < i)

\_putchar(' ');

}

\_putchar('\n');

}

}

}

9

#include "main.h"

#include <stdio.h>

/\*\*

\* main - prints the numbers from 1 to 100, followed by a new line

\* but for multiples of three prints Fizz instead of the number

\* and for the multiples of five prints Buzz

\* Return: Always 0 (Success)

\*/

int main(void)

{

int i;

for (i = 1; i <= 100; i++)

{

if (i % 3 == 0 && i % 5 != 0)

{

printf(" Fizz");

} else if (i % 5 == 0 && i % 3 != 0)

{

printf(" Buzz");

} else if (i % 3 == 0 && i % 5 == 0)

{

printf(" FizzBuzz");

} else if (i == 1)

{

printf("%d", i);

} else

{

printf(" %d", i);

}

}

printf("\n");

return (0);

}

10

#include "main.h"

/\*\*

\* print\_triangle - prints a triangle, followed by a new line

\* @size: size of the triangle

\*/

void print\_triangle(int size)

{

if (size <= 0)

{

\_putchar('\n');

}

else

{

int i, j;

for (i = 1; i <= size; i++)

{

for (j = i; j < size; j++)

{

\_putchar(' ');

}

for (j = 1; j <= i; j++)

{

\_putchar('#');

}

\_putchar('\n');

}

}

}

11

#include <stdio.h>

#include <math.h>

/\*\*

\* main - finds and prints the largest prime factor of the number 612852475143

\* followed by a new line

\* Return: Always 0 (Success)

\*/

int main(void)

{

long int n;

long int max;

long int i;

n = 612852475143;

max = -1;

while (n % 2 == 0)

{

max = 2;

n /= 2;

}

for (i = 3; i <= sqrt(n); i = i + 2)

{

while (n % i == 0)

{

max = i;

n = n / i;

}

}

if (n > 2)

max = n;

printf("%ld\n", max);

return (0);

}

13

#include "main.h"

/\*\*

\* print\_number - prints an integer

\* @n: integer to be printed

\*/

void print\_number(int n)

{

unsigned int n1;

if (n < 0)

{

n1 = -n;

\_putchar('-');

}

else

{

n1 = n;

}

if (n1 / 10)

{

print\_number(n1 / 10);

}

\_putchar((n1 % 10) + '0');

}

13