Python

1) C
2) B
3) C
4) A
5) D
6) C
7) A
8) C
9) A and C
10) A and B

```
{
 "cells": [
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "11. Python program to find the factorial of a number"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 2,
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Enter the number:6\n",
      "Factorial of 6 is 720\n"
     ]
    }
   ],
   "source": [
    "import math\n",
    " \n",
    "def factorial(n):\n",
    "    return(math.factorial(n))\n",
    "\n",
    "n=int(input(\"Enter the number:\"))\n",
    "print(\"Factorial of\", n, \"is\",factorial(n))"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "12.Python program to find whether a number is prime or composite"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 3,
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Enter the number3\n",
      "3 is a prime number\n"
     ]
    }
   ],
   "source": [
    "num = int(input(\"Enter the number:\"))\n",
    "  \n",
    "if num > 1:\n",
    "    for i in range(2, int(num/2)+1):\n",
    "        if (num % i) == 0:\n",
    "            print(num, \"is not a prime number\")\n",
    "            break\n",
    "    else:\n",
    "        print(num, \"is a prime number\")\n",
    "else:\n",
    "    print(num, \"is not a prime number\")"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "13.Python program to check whether a given string is palindrome or
not"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 4,
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Enter the string:madam\n",
      "Yes,It is a Palindrome\n"
     ]
    }
   ],
   "source": [
    "def isPalindrome(str):\n",
    "    for i in range(0, int(len(str)/2)):\n",
    "        if str[i] != str[len(str)-i-1]:\n",
    "            return False\n",
    "    return True\n",
    "s = str(input(\"Enter the string:\"))\n",
    "ans = isPalindrome(s)\n",
    " \n",
    "if (ans):\n",
    "    print(\"Yes,It is a Palindrome\")\n",
    "else:\n",
    "    print(\"No,It is not a Palindrome\")"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "14.Python program to get the third side of right-angled triangle
from two given sides"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 11,
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Enter the side 1:6\n",
      "Enter the side 2:8\n",
      "('Hypotenuse is:', 10.0)\n"
     ]
    }
   ],
   "source": [
    "def findHypotenuse(side1, side2):\n",
    " \n",
    "    h = (((side1 * side1) + (side2 * side2))**(1/2));\n",
    "    return \"Hypotenuse is:\",h;\n",
    "\n",
    "side1 = int(input(\"Enter the side 1:\"))\n",
    "side2 = int(input(\"Enter the side 2:\")) \n",
    " \n",
    "print(findHypotenuse(side1, side2));"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "15.Python program to print the frequency of each of the characters
present in a given string"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 13,
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Enter the string:bharath\n",
      "Frequency of each character :\n",
      "  {'b': 1}\n",
      "Frequency of each character :\n",
      "  {'b': 1, 'h': 1}\n",
      "Frequency of each character :\n",
      "  {'b': 1, 'h': 1, 'a': 1}\n",
      "Frequency of each character :\n",
      "  {'b': 1, 'h': 1, 'a': 1, 'r': 1}\n",
      "Frequency of each character :\n",
      "  {'b': 1, 'h': 1, 'a': 2, 'r': 1}\n",
      "Frequency of each character :\n",
      "  {'b': 1, 'h': 1, 'a': 2, 'r': 1, 't': 1}\n",
      "Frequency of each character :\n",
      "  {'b': 1, 'h': 2, 'a': 2, 'r': 1, 't': 1}\n"
     ]
    }
   ],
   "source": [
    "strZ = str(input(\"Enter the string:\"))\n",
    "res = {}\n",
    "\n",
    "for keys in strZ:\n",
    "    res[keys] = res.get(keys, 0) + 1\n",
    "    print(\"Frequency of each character :\\n \",res)"
   ]
  }
 ],
 "metadata": {
  "kernelspec": {
   "display_name": "Python 3",
   "language": "python",
   "name": "python3"
  },
  "language_info": {
   "codemirror_mode": {
    "name": "ipython",
    "version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.8.5"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 4
   }
```