```
{
 "cells": [
  {
   "cell_type": "code",
   "execution_count": 2,
   "metadata": {},
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Number of elements in array:10\n",
      "115.3\n",
      "195.5\n",
      "120.5\n",
      "110.2\n",
      "90.4\n",
      "105.6\n",
      "110.9\n",
      "116.3\n",
      "122.3\n",
      "125.4\n",
      "Mean / Average is: 121.24000000000001\n",
      "Median is: 115.8\n",
      "variance is: 7017.724\n",
      "std dev 83.7718568494217\n",
      "standardisation:\n",
      "-0.3681427290722982\n",
      "-0.1866975448343304\n",
      "-0.1317865022359979\n",
      "-0.12343047401451247\n",
      "-0.07090686805089026\n",
      "-0.05896968487733974\n",
      "-0.008833515548427498\n",
      "0.012653414163963414\n",
      "0.04965868200197015\n",
      "0.8864552224678619\n",
      "Min max normalisation\n",
      "-0.850618458610847\n",
      "-0.850618458610847\n",
      "-0.850618458610847\n",
      "-0.850618458610847\n",
      "-0.850618458610847\n",
      "-0.850618458610847\n",
      "-0.850618458610847\n",
      "-0.850618458610847\n",
      "-0.850618458610847\n",
      "-0.850618458610847\n",
      "mode 0.0\n"
     ]
    }
   ],
   "source": [
    "import math\n",
    "a=[]\n",
    "n=int(input(\"Number of elements in array:\"))\n",
    "for i in range(0,n):\n",
```

```
"     l=float(input())\n",
"     a.append(l)\n",
"\n",
"\n",
"\n",
"#mean     \n",
"get_sum = sum(a)\n",
"mean = get_sum / n\n",
"    \n",
"print(\"Mean / Average is: \" + str(mean))\n",
"\n",
"#median\n",
"\n",
"a.sort()\n",
"if n % 2 == 0:\n",
"       median1 = a[n//2]\n",
"       median2 = a[n//2 - 1]\n",
"       median = (median1 + median2)/2\n",
"else:\n",
"       median = a[n//2]\n",
"\n",
"print(\"Median is: \" + str(median))\n",
"\n",
"#standard deviation\n",
"vari=0.0\n",
"for i in a:\n",
"       z=i-mean\n",
"       vari=vari+(z**2)\n",
"print(\"variance is:\",vari)\n",
"\n",
"sd=math.sqrt(vari)\n",
"print(\"std dev\",sd)\n",
"\n",
"# mode of elements\n",
"\n",
"\n",
"print('standardisation:')\n",
"for i in a:\n",
"       std=(i-mean)/sd\n",
"       print(std)\n",
"\n",
"\n",
"print('Min max normalisation')\n",
"z=a[9]-a[0]\n",
"for i in a:\n",
"       y=(1-a[0])/z\n",
"       print(y)\n",
"       \n",
"c=0.0\n",
"modee=0.0\n",
"for i in range(0,10):\n",
"       for j in range(i+1,10):\n",
"              if a[i]==a[j]:\n",
"                     c=c+1\n",
"       \n",
"       if c>l:\n",
"              l=c\n",
```

```
        "              modee=a[i]\n",
        "if l==1:\n",
        "       print(\"There is no mode\")\n",
        "\n",
        "else:\n",
        "       print(\"mode\",modee)\n",
        "\n"
       ]
      },
      {
       "cell_type": "code",
       "execution_count": null,
       "metadata": {},
       "outputs": [],
       "source": []
      },
      {
       "cell_type": "code",
       "execution_count": null,
       "metadata": {},
       "outputs": [],
       "source": []
      }
     ],
     "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": "3.8.10"
      }
     },
     "nbformat": 4,
     "nbformat_minor": 4
    }
```