****

**Session: 2018-2019**

**Final Report**

**Lab at home Project**

**(The Project is based on C or C++ or Python or HTML or Matlab.)**

**Faculty Name:- Dr. Suman Mukherjee**

**Course Code:- PHY109**

**Topic:- Coding Program for Physics Application**

**Section:- K18PG**

**Submitted by:-**

**Name:- Ashwani Kumar**

**Roll No.:- RK18PGB41**

**Registration No.:- 11806645**

**Table Of Contents:-**

**1.** Introduction……………………………………………………………………….**3**

**2.** Objective……………………………………………………………………….....**3**

**3.** Color Mapping of Temperature scalar field using using Python…………………**4**

**4.** Refractive index profile of different types of optical fiber using Python….........**12**

**5.** Understanding of photoelectric effect………………...........................................**20**

**Introduction:-**

It is an based project specially design for practicing Programming. We have discussed various Problem related to Science and Technology. We have provided an overview of how the Computer language like C/C++/Python/ and to get known to Matlab was introduced to us. We have discussed the ways of making program on Computer and responded. We have to discuss the design of the programming lesson for some modules of Physics and application like Refractive Index, Photoelectric Effect, and Color Mapping and plotting temperature on Map and how they are incorporated into the Computer Programming.

**Objective:-**

Technology is in abundance in our everyday lives-tablets, smartphones, social media. It is becoming increasingly important to be able to best utilize these devices. This technology is also becoming a central core of our jobs. Programming is now an invaluable skill that employers are seeking in their new job applicants, especially in the fields of Science, Technology, Engineering, and Math (STEM). With technology being the forefront of out daily lives, STEM jobs will only increase. Because of this, it is now even more important to expose students to writing computer programs. Not only does it prepare the students for the new job market, it also teaches the students critical thinking skills that they otherwise may not learn through traditional high school curriculum. The critical thinking skills obtained through the logical process of writing code and computational methods will better equip the students as they enter college and careers. The goal of this initiative was to improve students’ critical thinking skills through the challenges of writing code. We as a students should attain minimum standards of physics knowledge in our introductory and upper level classes as demonstrated by solving problems that require an understanding and application of physics concepts. We also need to attain a sufficient knowledge of other sciences, mathematics, and technology that support our upper level courses and their application to our world. It provided us opportunities for original research experience, both experimental and theoretical, while modeling accepted research methods, laboratory skills, and ethics of the scientific community. This mainly occurs in our introductory and upper level classes, but it also require us to participate in other investigatory activities, i.e., Programming in Physics. It has an integrated experience in an industry setting, get supervised by physics faculty. As we as a Students are required to be able to communicate technical information in discplinary settings and completing courses in written and computer computation by making program having compulsary in our general education requirements. It has build on those courses with our physics curriculum, cumulating with our research requirement. Communication skills are required throughout the process, from the literature search to the preliminary plan for investigation to the technical write-up of results. As a Students, we should be able to achieve the following objectives which are related to specific programming concepts. Some of these concepts are applicable to all programming languages like C/C++/Python/ and to get introduce to Matlab. However, in general, these objectives address basic programming concepts that has provided the us with the appropriate set of tools to be able to write any program in the future.

Our outcomes include a mastery of:

1) domain knowledge in the fields of physics,

2) research methods and laboratory skills, and

3) communication skills including technical and communication.

**3.)Objective: Color Mapping of Temperature scalar field using using Python:-**

**Problem Statement:** Draw a map of temperature of several neighboring cities of different colors. There must be input for temperature. we can choose to preset the name of the cities.

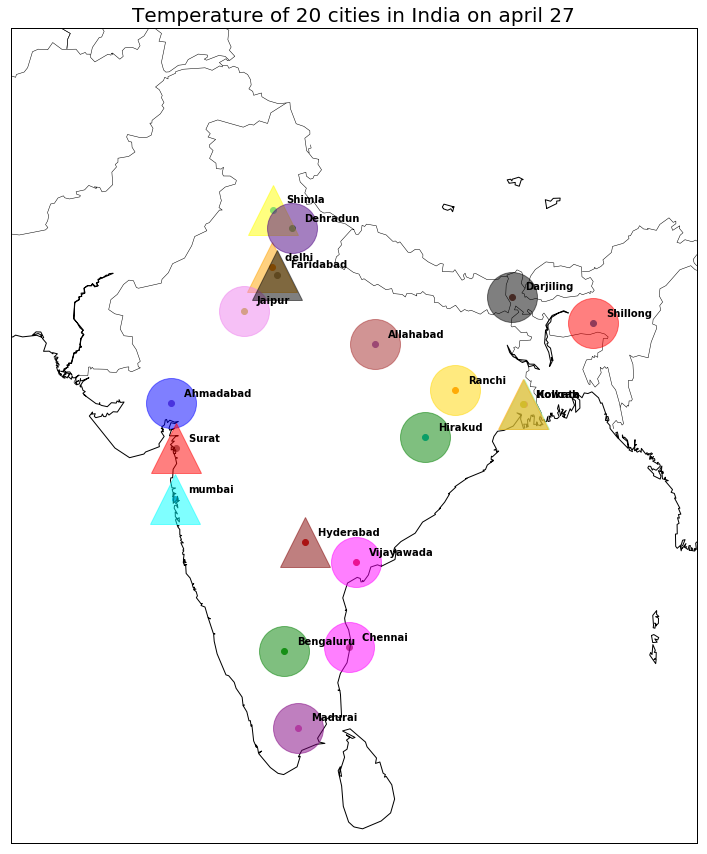
**Task done at home:** Taken 20 neighbouring cities/places and written down its temperature at a given day. Now, written a code using any software i wish to color map the temperature of those places with the name of the places on the map and had create an input to enter the temperature.

**Task done at tutorial:** to draw a color map in two dimensions taking several input data.

**Color Mapping of Temperature scalar field using Python**

|  |  |  |
| --- | --- | --- |
| **S.no** | **City** | **Temperature** |
| **1** | **Mumbai** | **30°C** |
| **2** | **Delhi** | **45°C** |
| **3** | **Bengaluru** | **35°C** |
| **4** | **Hyderabad** | **42°C** |
| **5** | **Ahmadabad** | **31°C** |
| **6** | **Chennai** | **38°C** |
| **7** | **Kolkata** | **33°C** |
| **8** | **Surat** | **49°C** |
| **9** | **Jaipur** | **43°C** |
| **10** | **Shimla** | **48°C** |
| **11** | **Shillong** | **49°C** |
| **12** | **Ranchi** | **47°C** |
| **13** | **Dehradun** | **36°C** |
| **14** | **Vijayawada** | **38°C** |
| **15** | **Allahabad** | **41°C** |
| **16** | **Darjiling** | **39°C** |
| **17** | **Madurai** | **37°C** |
| **18** | **Faridabad** | **39°C** |
| **19** | **Howrah** | **45°C** |
| **20** | **Hirakud** | **35°C** |

|  |  |  |
| --- | --- | --- |
| **S.no** | **Temperature** | **Colour** |
| **1** | **30°C** | **Cyan** |
| **2** | **31°C** | **Blue** |
| **3** | **32°C** | **Pink** |
| **4** | **33°C** | **Light green** |
| **5** | **34°C** | **White** |
| **6** | **35°C** | **Green** |
| **7** | **36°C** | **Indigo** |
| **8** | **37°C** | **Purple** |
| **9** | **38°C** | **Magenta** |
| **10** | **39°C** | **Black** |
| **11** | **40°C** | **Grey** |
| **12** | **41°C** | **Brown** |
| **13** | **42°C** | **Maroon** |
| **14** | **43°C** | **Violet** |
| **15** | **44°C** | **Deep pink** |
| **16** | **45°C** | **Orange** |
| **17** | **46°C** | **Dark orange** |
| **18** | **47°C** | **Gold** |
| **19** | **48°C** | **Yellow** |
| **20** | **49°C** | **Red** |

****

**Code:**

**import matplotlib.pyplot as plt**

**from mpl\_toolkits.basemap import Basemap**

**plt.subplots(figsize=(20, 15))**

**map = Basemap(width=1200000,height=900000,projection='lcc',resolution='l',**

**llcrnrlon=67,llcrnrlat=5,urcrnrlon=99,urcrnrlat=37,lat\_0=28,lon\_0=77)**

**map.drawmapboundary ()**

**map.drawcountries ()**

**map.drawcoastlines ()**

**lg=[72.8777]**

**lt=[19.0760]**

**nc=['mumbai']**

**x, y = map(lg, lt)**

**plt.scatter(x, y, marker="o")**

**map.plot(x,y,'k^',markersize=50,alpha=0.5,color='cyan')**

**for ncs, xpt, ypt in zip(nc, x, y):**

**plt.text(xpt+60000, ypt+30000, ncs, fontsize=10, fontweight='bold')**

**lk=[77.1025]**

**li=[28.7041]**

**mn=['delhi ']**

**x,y=map(lk,li)**

**plt.scatter(x, y, marker="o")**

**map.plot(x,y,'k^',markersize=50,alpha=0.5,color='orange')**

**for mns, xpt, ypt in zip(mn, x, y):**

**plt.text(xpt+60000, ypt+30000, mns, fontsize=10, fontweight='bold')**

**la=[77.5946]**

**lb=[12.9716]**

**na=['Bengaluru']**

**x, y = map(la, lb)**

**plt.scatter(x, y, marker="o")**

**map.plot(x,y,'ko',markersize=50,alpha=0.5,color='green')**

**for nas, xpt, ypt in zip(na, x, y):**

**plt.text(xpt+60000, ypt+30000, nas, fontsize=10, fontweight='bold')**

**lc=[78.4867]**

**ld=[17.3850]**

**nb=['Hyderabad ']**

**x, y = map(lc, ld)**

**plt.scatter(x, y, marker="o")**

**map.plot(x,y,'k^',markersize=50,alpha=0.5,color='maroon')**

**for nbs, xpt, ypt in zip(nb, x, y):**

**plt.text(xpt+60000, ypt+30000, nbs, fontsize=10, fontweight='bold')**

**lf=[72.5713621]**

**lh=[23.022505]**

**nd=['Ahmadabad ']**

**x, y = map(lf, lh)**

**plt.scatter(x, y, marker="o")**

**map.plot(x,y,'ko',markersize=50,alpha=0.5,color='blue')**

**for nds, xpt, ypt in zip(nd, x, y):**

**plt.text(xpt+60000, ypt+30000, nds, fontsize=10, fontweight='bold')**

**li=[80.2707184]**

**lj=[13.0826802]**

**ne=['Chennai ']**

**x, y = map(li, lj)**

**plt.scatter(x, y, marker="o")**

**map.plot(x,y,'ko',markersize=50,alpha=0.5,color='magenta')**

**for nes, xpt, ypt in zip(ne, x, y):**

**plt.text(xpt+60000, ypt+30000, nes, fontsize=10, fontweight='bold')**

**lk=[88.363895]**

**ll=[22.572646]**

**nf=['Kolkata ']**

**x, y = map(lk, ll)**

**plt.scatter(x, y, marker="o")**

**map.plot(x,y,'k^',markersize=50,alpha=0.5,color='lightgreen')**

**for nfs, xpt, ypt in zip(nf, x, y):**

**plt.text(xpt+60000, ypt+30000, nfs, fontsize=10, fontweight='bold')**

**lm=[72.8310607]**

**ln=[21.1702401]**

**ng=['Surat ']**

**x, y = map(lm, ln)**

**plt.scatter(x, y, marker="o")**

**map.plot(x,y,'k^',markersize=50,alpha=0.5,color='red')**

**for ngs, xpt, ypt in zip(ng, x, y):**

**plt.text(xpt+60000, ypt+30000, ngs, fontsize=10, fontweight='bold')**

**lr=[75.7872709]**

**ls=[26.9124336]**

**ni=['Jaipur ']**

**x, y = map(lr, ls)**

**plt.scatter(x, y, marker="o")**

**map.plot(x,y,'ko',markersize=50,alpha=0.5,color='violet')**

**for nis, xpt, ypt in zip(ni, x, y):**

**plt.text(xpt+60000, ypt+30000, nis, fontsize=10, fontweight='bold')**

**lu=[77.16662]**

**lv=[31.10442]**

**nk=['Shimla']**

**x, y = map(lu, lv)**

**plt.scatter(x, y, marker="o")**

**map.plot(x,y,'k^',markersize=50,alpha=0.5,color='yellow')**

**for nks, xpt, ypt in zip(nk, x, y):**

**plt.text(xpt+60000, ypt+30000, nks, fontsize=10, fontweight='bold')**

**lx=[91.88313]**

**lz=[25.56892]**

**nj=['Shillong']**

**x, y = map(lx, lz)**

**plt.scatter(x, y, marker="o")**

**map.plot(x,y,'ko',markersize=50,alpha=0.5,color='red')**

**for njs, xpt, ypt in zip(nj, x, y):**

**plt.text(xpt+60000, ypt+30000, njs, fontsize=10, fontweight='bold')**

**e=[85.33856]**

**f=[23.34777]**

**nl=['Ranchi']**

**x, y = map(e, f)**

**plt.scatter(x, y, marker="o")**

**map.plot(x,y,'ko',markersize=50,alpha=0.5,color='gold')**

**for nls, xpt, ypt in zip(nl, x, y):**

**plt.text(xpt+60000, ypt+30000, nls, fontsize=10, fontweight='bold')**

**g=[78.03392]**

**h=[30.32443]**

**nm=['Dehradun']**

**x, y = map(g, h)**

**plt.scatter(x, y, marker="o")**

**map.plot(x,y,'ko',markersize=50,alpha=0.5,color='indigo')**

**for nms, xpt, ypt in zip(nm, x, y):**

**plt.text(xpt+60000, ypt+30000, nms, fontsize=10, fontweight='bold')**

**i=[ 80.63049]**

**j=[16.51928]**

**nn=['Vijayawada']**

**x, y = map(i, j)**

**plt.scatter(x, y, marker="o")**

**map.plot(x,y,'ko',markersize=50,alpha=0.5,color='magenta')**

**for nns, xpt, ypt in zip(nn, x, y):**

**plt.text(xpt+60000, ypt+30000, nns, fontsize=10, fontweight='bold')**

**k=[81.83329]**

**l=[25.44894]**

**no=['Allahabad']**

**x, y = map(k, l)**

**plt.scatter(x, y, marker="o")**

**map.plot(x,y,'ko',markersize=50,alpha=0.5,color='brown')**

**for nos, xpt, ypt in zip(no, x, y):**

**plt.text(xpt+60000, ypt+30000, nos, fontsize=10, fontweight='bold')**

**m=[88.26667]**

**n=[27.03333]**

**np=['Darjiling']**

**x, y = map(m, n)**

**plt.scatter(x, y, marker="o")**

**map.plot(x,y,'ko',markersize=50,alpha=0.5,color='black')**

**for nps, xpt, ypt in zip(np, x, y):**

**plt.text(xpt+60000, ypt+30000, nps, fontsize=10, fontweight='bold')**

**o=[78.11962]**

**p=[9.91735]**

**nq=['Madurai']**

**x, y = map(o, p)**

**plt.scatter(x, y, marker="o")**

**map.plot(x,y,'ko',markersize=50,alpha=0.5,color='purple')**

**for nqs, xpt, ypt in zip(nq, x, y):**

**plt.text(xpt+60000, ypt+30000, nqs, fontsize=10, fontweight='bold')**

**q=[77.31977]**

**r=[28.41252]**

**nr=['Faridabad']**

**x, y = map(q, r)**

**plt.scatter(x, y, marker="o")**

**map.plot(x,y,'k^',markersize=50,alpha=0.5,color='black')**

**for nrs, xpt, ypt in zip(nr, x, y):**

**plt.text(xpt+60000, ypt+30000, nrs, fontsize=10, fontweight='bold')**

**s=[88.31857]**

**t=[22.57688]**

**nt=['Howrah ']**

**x, y = map(s, t)**

**plt.scatter(x, y, marker="o")**

**map.plot(x,y,'k^',markersize=50,alpha=0.5,color='orange')**

**for nts, xpt, ypt in zip(nt, x, y):**

**plt.text(xpt+60000, ypt+30000, nts, fontsize=10, fontweight='bold')**

**u=[83.86667]**

**v=[21.51667]**

**nu=['Hirakud']**

**x, y = map(u, v)**

**plt.scatter(x, y, marker="o")**

**map.plot(x,y,'ko',markersize=50,alpha=0.5,color='green')**

**for nus, xpt, ypt in zip(nu, x, y):**

**plt.text(xpt+60000, ypt+30000, nus, fontsize=10, fontweight='bold')**

**plt.title('Temperature of 20 cities in India on april 27',fontsize=20)**

**4.) Objective: Refractive index profile of different types of optical fiber using Python:-**

**Problem Statement:** Draw refractive index profile (Change of refractive index with radial distance) of different types of optical fibre by using your own input data.

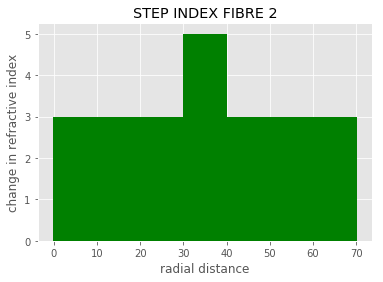
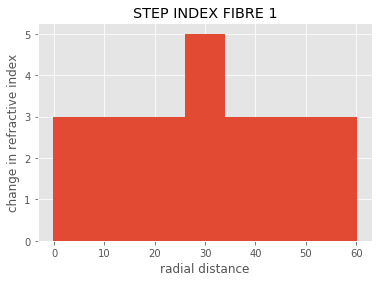
**Task done at home:** Taken at last four different optical fibre, two step index and two graded index fibre. I have defined their core and cladding diameter and done the coding at home to draw the refractive index profile.

**Task done at tutorial:** I have define the function of refractive index dependent on radial distance for graded index fibre

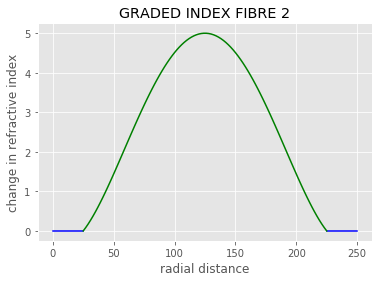
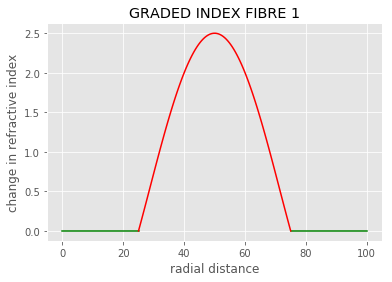
**Program for Plotting Graph for Refractive index profile of different types of optical fiber using Python:-**

**4. Refractive index profile of different types of optical fiber**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.no** | **Optical fiber** | **Core Diameter** | **Cladding Diameter** |
| **1** | **Step index fiber 1** | **8μm** | **60μm** |
| **2** | **Step index fiber 2** | **10μm** | **70μm** |

****

|  |  |  |  |
| --- | --- | --- | --- |
| **S.no** | **Optical fiber** | **Core Diameter** | **Cladding Diameter** |
| **1** | **Graded index fiber 1** | **50μm** | **100μm** |
| **2** | **Graded index fiber 2** | **200μm** | **250μm** |

****

**Code:**

**from matplotlib import pyplot as plt**

**from matplotlib import style**

**style.use('ggplot')**

**x = [0,10,26,30,35,60]**

**y = [3,3,3,5,3,3]**

**width = [0.5,20,20,8,50,0.4]**

**plt.bar(x, y,width, align='center')**

**plt.ylabel('change in refractive index')**

**plt.xlabel('radial distance')**

**plt.title('STEP INDEX FIBRE 1')**

**plt.show()**

**print("Core Diameter= 8μm")**

**print("Cladding Diameter =60μm")**

**x2 = [0,10,20,30,35,40,50,60,70]**

**y2 = [3,3,3,3,5,3,3,3,3]**

**width = [0.5,20,10,10,10,10,10,20,0.5]**

**plt.bar(x2, y2,width,color='g',align='center')**

**plt.ylabel('change in refractive index')**

**plt.xlabel('radial distance')**

**plt.title('STEP INDEX FIBRE 2')**

**plt.show()**

**print("Core Diameter= 10μm")**

**print("Cladding Diameter =70μm")**

**import numpy as np**

**import scipy.interpolate as interpolate**

**import matplotlib.pyplot as plt**

**x = [0,10,15,20,25]**

**y = [0,0,0,0,0]**

**x2 = [75,85,90,95,100]**

**y2 = [0,0,0,0,0]**

**plt.plot(x,y,color='g')**

**plt.plot(x2,y2,color='g')**

**x = np.array([25,40,50,60,75 ])**

**y = np.array([0,2,2.5,2,0])**

**t, c, k = interpolate.splrep(x, y, s=0, k=4)**

**N = 100**

**xmin, xmax = x.min(), x.max()**

**xx = np.linspace(xmin, xmax, N)**

**spline = interpolate.BSpline(t, c, k, extrapolate=False)**

**plt.plot(xx, spline(xx), 'r')**

**plt.title('GRADED INDEX FIBRE 1')**

**plt.ylabel('change in refractive index')**

**plt.xlabel('radial distance')**

**plt.show()**

**print("Core Diameter= 50μm")**

**print("Cladding Diameter =100μm")**

**x = [0,10,15,20,25]**

**y = [0,0,0,0,0]**

**x2 = [225,235,240,245,250]**

**y2 = [0,0,0,0,0]**

**plt.plot(x,y,color='b')**

**plt.plot(x2,y2,color='b')**

**x = np.array([25,65,125,185,225])**

**y = np.array([0,2.5,5,2.5,0])**

**t, c, k = interpolate.splrep(x, y, s=0, k=4)**

**N = 100**

**xmin, xmax = x.min(), x.max()**

**xx = np.linspace(xmin, xmax, N)**

**spline = interpolate.BSpline(t, c, k, extrapolate=False)**

**plt.plot(xx, spline(xx), 'g')**

**plt.title('GRADED INDEX FIBRE 2')**

**plt.ylabel('change in refractive index')**

**plt.xlabel('radial distance')**

**plt.show()**

**print("Core Diameter= 200μm")**

**print("Cladding Diameter =250μm")**

**Another Program for Plotting Graph for Refractive index profile of different types of optical fiber using Python:-**

**( Note:- Program is done on Python )**

**First program for Step Index Fibre:-**

**In [106]:**

**import matplotlib.pyplot as plot**

**Step index optical ﬁber**

**In [107]:**

**radius = 50 # radius of core**

**n1 = 1.5 # core**

**n2 = 1.2 # cladding**

**In [108]:**

**distance = [x for x in range(-100, 100)]**

**In [109]:**

**def calculate\_y(distance):**

**y = list()**

**for x in distance:**

**if x > -50 and x < 50:**

**y.append(n1)**

**else:**

**y.append(n2)**

**return y**

**In [120]:**

**y = calculate\_y(distance)**

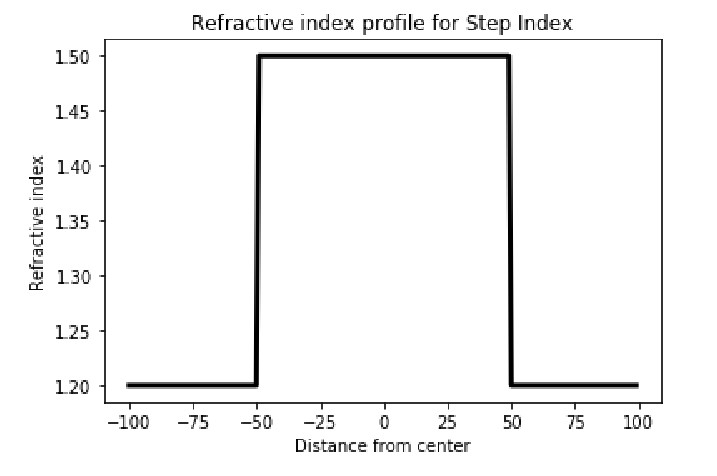
**mlt.plot(distance, y, 'k', linewidth=3)**

**mlt.xlabel('Distance from center')**

**mlt.ylabel('Refractive index')**

**mlt.title('Refractive index profile for Step Index')**

**mlt.show()**

****

**Second program for Graded Index Fibre (Triangular):-**

**Graded Index Fiber optical ﬁber**

**In [132]: n1 = 1.5 # core**

**n2 = 1.1 # cladding**

**radius = 50 # radius of core**

**a = 1 # alpha for triangular profile**

**relative\_index = (n1-n2)\*2/(n1+n2)**

**In [133]:**

**distance = [x for x in range(-100, 100)]**

**In [134]:**

**def calculate\_y(distance):**

**y = list()**

**for x in distance:**

**if x > -radius and x < radius:**

**y.append(n1\*((1-2\*relative\_index\*((x/radius)\*\*a))\*\*1/2))**

**else:**

**y.append(n1\*((1-2\*relative\_index)\*\*1/2))**

**return y**

**In [135]:**

**y = calculate\_y(distance)**

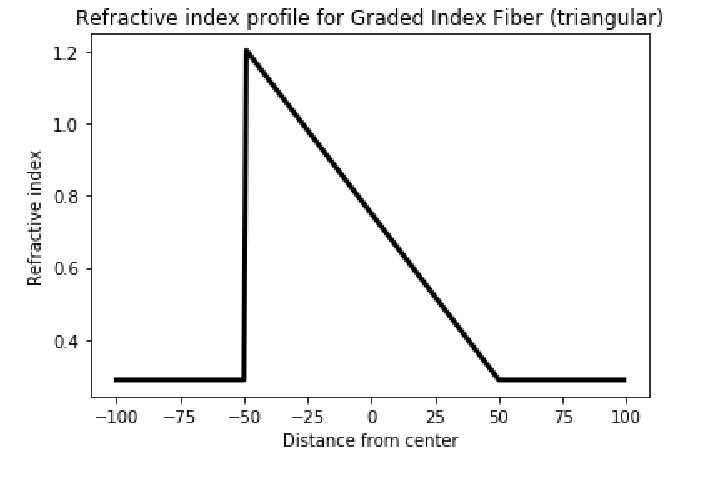
**mlt.plot(distance, y, 'k', linewidth=3)**

**mlt.xlabel('Distance from center')**

**mlt.ylabel('Refractive index')**

**mlt.title('Refractive index profile for Graded Index Fiber (triangular)')**

**mlt.show()**

****

**Third program for Graded Index Fibre (Parabolic):-**

**In [136]:**

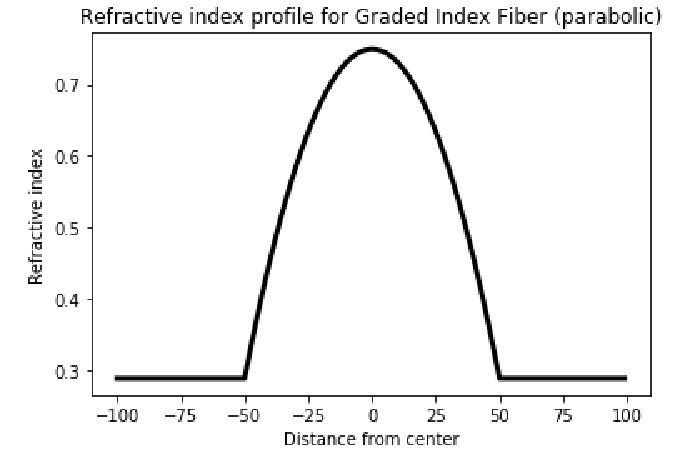
**a = 2 # alpha for parabolic profile**

**y = calculate\_y(distance) mlt.plot(distance, y,'k', linewidth=3)**

**mlt.xlabel('Distance from center') mlt.ylabel('Refractive index')**

**mlt.title('Refractive index profile for Graded Index Fiber (parabolic)')**

**mlt.show()**

****

**In [ ]:**

**Conclusion:**

**Step index fiber:**

**The RI is constant for the core in this fiber. As we go radially from center of the core, the RI undergoes a step change at core-cladding interface.**

**Graded index fiber:**

**In this fiber , the RI of Core continuously decreases from center to the surface (radially).**

**The RI is maximum at the center of Core and Minimum at the Surface.**

**5.) Objective: Understanding of photoelectric effect:-**

**Problem Statement:** Use <https://phet.colorado.edu/en/simulation/photoelectric> to understand the phenomena of photoelectric effect.

**Task to be done at home:** Plot light energy vs. kinetic energy of the emitted electrons and Write Python code to find work function, threshold frequency.

**Task to be done at tutorial:** Practice writing codes to draw graph and how to work with function.

**Program for Caculating Work Function and Threshold Frequency : -**

**( Note:- Program done in C Language )**

**#include<stdio.h>**

**#include<math.h>**

**#include<stdlib.h>**

**#include<conio.h>**

**void main()**

**{**

**int s,e,pr;**

**char ch;**

**double lo,wo,nm;**

**double a,vo,h,c,w;**

**double p1,p2,p3,p4,p5,p6,p7;**

**printf("----------------------------------------------------------------------------------------------\n");**

**printf("----------------------------------------------------------------------------------------------\n");**

**printf(" SUBMITTED TO : Dr. SUMAN MUKHERJEE \n");**

**printf(" SUBMITTED BY :- \n ");**

**printf(" Reg No.:- 11806645 NAME : Ashwani Kumar ROLL: B-41 SECTION : K18PG GROUP- 2 :\n");**

**printf("----------------------------------------------------------------------------------------------\n");**

**printf("----------------------------------------------------------------------------------------------\n");**

**printf("This program is going to calculate : WORK FUNCTION and THRESHOLD FREQUECY\n ");**

**printf("----------------------------------------------------------------------------------------------\n");**

**printf("press any key to enter into the main program");**

**ch=getch();**

**system("cls");**

**printf("----------------------------------------------------------------------------------------------\n");**

**case1:**

**printf("\nplease select any one of the following in which units you want to give for threshold wavelength(vo)\n");**

**printf("1)nano metre nm\n2)Angstrom(Ao)\n");**

**printf("1)nm\n2)Ao\n");**

**printf("enter 1. for nano metre(nm)\n");**

**printf("enter 2. for Angstrom(Ao)\n");**

**scanf("%d",&s);**

**printf("............................\n");**

**if(s==1)**

**{**

**system("cls");**

**printf("\nyou have selected\n1) nano metre");**

**printf("\nnow please give input of Threshold wavelength lambda\"Lo\" in nano metres(nm) Lo = ");**

**scanf("%lf",&lo);**

**wo=1242.375/lo;**

**printf("\nWork Functoin (Wo)= %.6lf ev",wo);**

**vo=wo\*2413273;**

**printf("\n\nThreshold Frequency(Vo)=%.6lf x 1e+5 Hz",vo);**

**}**

**else if(s==2)**

**{**

**system("cls");**

**printf("\nyou have selected\n2) Angstrom");**

**printf("\nnow please give input of Threshold wavelength lambda Lo in Angstrom(a) Lo = ");**

**scanf("%lf",&lo);**

**wo=12423.75/lo;**

**printf("\nWork Functoin (Wo)= %.6lf ev",wo);**

**vo=wo\*2413273;**

**printf("\n\nThreshold Frequency(Vo)=%.6lf x 1e+5 Hz",vo);**

**}**

**else{**

**printf("1)nano metre nm\n2)Angstrom(Ao)\n");**

**printf("please enter only 1 or 2");**

**goto case1;**

**}**

**printf("\n----------------------------------------------------------------------------------------------\n");**

**printf("\n----------------------------------------------------------------------------------------------\n");**

**printf("\ndo you want to continue");**

**printf("\npress 1 to continue");**

**printf("\nor ");**

**printf("\npress 0 to stop\n");**

**char p;**

**scanf("%d",&p);**

**if(p==1)**

**{**

**printf("............................\n");**

**system("cls");**

**goto case1;**

**}**

**}**

**Another Program on Python:-**

**from matplotlib import pyplot as plt**

**x = [1,2,3,4]**

**y = [3,3,3,3]**

**plt.plot(x,y)**

**plt.title('Light Intensity vs. Kinetic Energy ')**

**plt.ylabel('Kinetic Energy')**

**plt.xlabel('Light Intensity')**

**plt.show()**

**print("plancks constant =6.6^10-34")**

**h =6.624\*10\*\*-34**

**c=3\*10\*\*8**

**print("enter frequency ")**

**f=float(input())**

**#print("enter kinetic energy")**

**#k=float(input())**

**#print("enter wavelength in nm")**

**#l=float(input())**

**print("enter mass of body")**

**m=int(input())**

**print("enter velocity of paerticle")**

**v=int(input())**

**w=-(h\*f-0.5\*m\*v\*v)**

**#w=-((h\*c)/(l\*10\*\*-9)-k)**

**print("work function =",w,"ev")**

**tf=(w\*1.6\*10\*\*-19)/h**

**print("threshold frequency=",tf,"Hz")**

**Program for Plotting Graph for Kinetic Energy (K.E) vs Light Intensity (L.I):-**

**( Note:- Program is done on Python )**

**import matplotlib.pyplot as plt**

**from matplotlib import style**

**from numpy import \***

**#style.use('dark\_background')**

**print(plt.style.available)**

**plt.plot([1,2,3],[3,3,3])**

**plt.xlabel('KINETIC ENERGY')**

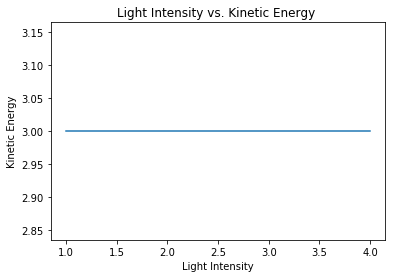
**plt.ylabel('LIGHT INTENSITY')**

**plt.title('INTENSITY VS K.E')**

**plt.legend()**

**plt.grid(True)**

**plt.show()**

****

**Conclusion:**

**By using the above code we can calculate work function and threshold frequency.**

**For increasing values of kinetic energy or light intensity we can see that they are constant to each other.**