**Assignment-1(Day 1)**

**1)** After running the following code, what does the variable bacon contain?

bacon = 22

bacon + 1

**Ans:-** a)If the bacon is printed then the answer will be **22**

b)if the bacon is assigned to any other variable or same variable or if

print(bacon+1) is printed then answer will be **23**

**2)** What should the values of the following two terms be?

'spam' + 'spamspam'

'spam' \* 3

**Ans:-** the answer for both the statement will be **spamspamspam.**

**3)** How can you tell the difference between break and continue?

**Ans:-**

**a)break:** break statement terminates the loop. Control of the program flows to the statement next loop. If the break statement is inside a nested loop the break statement will terminate the innermost loop.

**b)continue:** continue statement skips a single iteration in a loop.

Both break and continue statements can be used in a for or a while loop

**4)** In a for loop, what is the difference between range(10), range(0, 10), and range(0, 10, 1)?

**Ans:-**

range(start,stop,step)

start: An integer number specifying at which position to start.

stop: An integer number specifying at which position to stop(not included).

Step: An integer number specifying the incrementation.

For the given question the answer for the three condition will be same that is:

0

1

2

3

4

5

6

7

8

9

**5)** Using a for loop, write a short programme that prints the numbers 1 to 10 Then, using a while loop, create an identical programme that prints the numbers 1 to 10.

**Ans:-**

**Program:**

for i in range(1,11):

print(i)

print('\n')

a=1

while(a<=10):

print(a)

a=a+1

**Output:**

1

2

3

4

5

6

7

8

9

10

1

2

3

4

5

6

7

8

9

10

**6)** Given a number x, determine whether the given number is Armstrong number or not.

Input : 153

Output : Yes

153 is an Armstrong number.

1 \* 1 \* 1 + 5 \* 5 \* 5 + 3 \* 3 \* 3 = 153

**Program:**

num=153

r=len(str(num))

sum = 0

temp = num

while temp > 0:

digit = temp % 10

sum += digit \*\*r

temp //= 10

if num == sum:

print("yes")

print(num,"is an Armstrong number")

else:

print("no")

print(num,"is not an Armstrong number")

**Output:**

yes

153 is an Armstrong number

**7)**Program to find Sum of squares of first n natural numbers.

**Program:**

n=2

sum=0

for i in range(1,n+1):

sum=sum+(i\*i)

print(sum)

**Output:**

5

**8)**Program to Reverse words in a given String in Python.

**Program:**

str1= "I'm learning python and data science zero to hero"

words = str1.split()

words = list(reversed(words))

print(" ".join(words))

**Output:**

hero to zero science data and python learning I'm

**9)** Given a list of numbers, write a Python program to find the sum of all the elements in the list.

Input: [10,12,13]

Output: 35

**Program:**

l1=[10,12,13]

sum=0

a=0

while(a<len(l1)):

sum=sum+l1[a]

a=a+1

print("sum of the elements in the given list:",sum)

**Output:**

sum of the elements in the given list: 35

**10)** Write a Python program to print all even numbers between 10-1000.

**Program:**

for i in range(10,1001):

if i%2==0:

print(i,sep='')

**Output:**

10

12

14

16

18

20

22

24

26

28

30

32

34

36

38

40

42

44

46

48

50

52

54

56

58

60

62

64

66

68

70

72

74

76

78

80

82

84

86

88

90

92

94

96

98

100

102

104

106

108

110

112

114

116

118

120

122

124

126

128

130

132

134

136

138

140

142

144

146

148

150

152

154

156

158

160

162

164

166

168

170

172

174

176

178

180

182

184

186

188

190

192

194

196

198

200

202

204

206

208

210

212

214

216

218

220

222

224

226

228

230

232

234

236

238

240

242

244

246

248

250

252

254

256

258

260

262

264

266

268

270

272

274

276

278

280

282

284

286

288

290

292

294

296

298

300

302

304

306

308

310

312

314

316

318

320

322

324

326

328

330

332

334

336

338

340

342

344

346

348

350

352

354

356

358

360

362

364

366

368

370

372

374

376

378

380

382

384

386

388

390

392

394

396

398

400

402

404

406

408

410

412

414

416

418

420

422

424

426

428

430

432

434

436

438

440

442

444

446

448

450

452

454

456

458

460

462

464

466

468

470

472

474

476

478

480

482

484

486

488

490

492

494

496

498

500

502

504

506

508

510

512

514

516

518

520

522

524

526

528

530

532

534

536

538

540

542

544

546

548

550

552

554

556

558

560

562

564

566

568

570

572

574

576

578

580

582

584

586

588

590

592

594

596

598

600

602

604

606

608

610

612

614

616

618

620

622

624

626

628

630

632

634

636

638

640

642

644

646

648

650

652

654

656

658

660

662

664

666

668

670

672

674

676

678

680

682

684

686

688

690

692

694

696

698

700

702

704

706

708

710

712

714

716

718

720

722

724

726

728

730

732

734

736

738

740

742

744

746

748

750

752

754

756

758

760

762

764

766

768

770

772

774

776

778

780

782

784

786

788

790

792

794

796

798

800

802

804

806

808

810

812

814

816

818

820

822

824

826

828

830

832

834

836

838

840

842

844

846

848

850

852

854

856

858

860

862

864

866

868

870

872

874

876

878

880

882

884

886

888

890

892

894

896

898

900

902

904

906

908

910

912

914

916

918

920

922

924

926

928

930

932

934

936

938

940

942

944

946

948

950

952

954

956

958

960

962

964

966

968

970

972

974

976

978

980

982

984

986

988

990

992

994

996

998

1000