1.

   If **following** program having name cmdline is run from the command line as

   "./cmdline.out PG-DBDA PG-DESD PG-DMC PG-DAC PG-DITISS"
then what would be the output?

```c
#include<stdio.h>
int main(int argc, char *argv[])
{
    int i=1, len=argc;
    while(argv[i])
    {
        printf("%c",argv[i++][argc-len]);
    }
    return 0;
}
```

A. PPPPP
B. PG-DAT
C. ADCCS
D. PPPP
E. Run time error

Answer: A

2.

   What **will** be the output of following program **if** base address of arr is **4289999264.**   **Noe** - consider **64** bit compiler

```c
#include<stdio.h>
int main(void)
{
    int a[2][2] = {{1,2},{3,4}};
    printf("%u ", &a[0]);
    printf("%u ", &a+1);
    printf("%u ", a+1);
    printf("%u \n", &(a+1));
    return 0;
}
```

A. **4289999264 4289999280 4289999272 4289999280**
B. **4289999264 4289999272 4289999268 4289999280**
C. **4289999264 4289999272 4289999280 4289999272**
D. **Compile time error**
E. **Run time error**

Answer: D

3.

```c
#include<stdio.h>
int main(void)
{
    static char *s[] ={ "PG-DAC" , "PG-DITISS" ,"PG-DMC",
                        "PG-DBDA", "PG-DESD"};
    char **ptr[] = {s+4, s+3, s+2, s+1, s};
    char ***p=NULL;

    p = &ptr[2];
    p--;
    --p;
    printf("%s",(**p));

    return 0;
}
```

A. **DMC**
B. **PG-DBDA**
C. **PG-DESD**
D. **DITISS**
E. **PG-DITISS**

Answer: C

4.

```c
#include<stdio.h>
void funcion_print(int p[][2])
{
    int r,c;
    for(r=0; r<2; r++)
        for(c=0; c<2; c++)
        {
            printf("%d %d %d %d ",r,c,*(*(p+r)+c),*(*(p+c)+r));
            printf("%d %d %d %d \n",c,r,p[c][r], p[r][c]);
        }
    return;
}
int main(void)
{
    int a[2][2] = {{1,2},{1,2}};
    funcion_print(a);
    return 0;
}
```

A. 0 0 1 1 0 0 1 1
   0 1 2 1 1 0 1 2
   1 0 1 2 0 1 2 1
   1 1 2 2 1 1 2 2
B. 0 0 1 1 0 0 1 1
   0 1 1 2 1 0 2 1
   1 0 2 1 0 1 1 2
   1 1 2 2 1 1 2 1
C. 0 0 1 1 0 0 1 1
   0 1 2 2 1 0 2 2
   1 0 1 1 0 1 1 1
   1 1 2 2 1 1 2 2
D. 0 0 1 1 0 0 1 1
   0 1 1 1 0 1 1 1
   1 0 2 2 0 1 2 2
   1 1 2 2 1 1 2 2

Answer: A

5.

```c
#include<stdio.h>
int cnt;
void fun(int **const pp);
int main(void)
{
    int arr[][4]={{1,2,3,4},{4,3,2,8},{7,8,9,0}}, r,c;
    int *ptr=NULL;
    for(r=0; r<3; r++)
    {
        for( c=0;c<4; c++)
        {
            ptr = &arr[r][c];
            fun(&ptr);
            printf("%d ", arr[r][c]);
        }
    }
    printf("\n");
    return 0;
}
void fun(int **const pp)
{
    if( cnt++%2==0)
        **pp= **pp + sizeof('A')/2;
    else
        **pp= **pp + sizeof(3.142)/2;
    return;
}
```

A. 3 6 5 8 6 7 4 12 9 12 11 4
B. 5 4 7 6 8 5 6 10 11 10 13 2
C. 3 4 5 6 6 5 4 10 9 10 11 2
D. 3 6 5 8 7 6 4 12 9 12 11 4
E. 1 2 3 4 4 3 2 8 7 8 9 0
F. Error

Answer: A