1 - E) Entrará em looping infinito.

2 - Trecho 1: Irá imprimir “1 2 3 4”
Trecho 2: Irá imprimir “1 2 3 4” da mesma forma, pois a matriz é alocada sequencialmente e o ponteiro está fazendo o acesso de forma sequencial.

3 - a) P aponta para i que tem valor 3;
b) P aponta para i (3) e Q aponta para j (5) então é o mesmo que 3 - 5, resultando em -2;
c) **&p é o mesmo que *p, pois o primeiro asterísco se cancela com o &, ou seja, essa expressão irá retornar o valor no endereço *p, que é 3.
d) Provavelmente a resposta é uma demissão, 3* - *p não deveria ser uma operação válida, mas o que parece o compilador não se importa, teoricamente a sintaxe que um programador não psicopata escreveria seria algo como:
((3 * (- *p)) / (*q)) +7
mas enfim, a resposta pra isso seria 6.

4 - A saída seria: “endereço 1000”, 7, 5, 15, 9;

5 - A) *(pulo + 2)

6 - a) p= mat + 1; É válido, pois está acessando o segundo elemento do vetor;
b) p= mat++; Inválido, pois mat++ é o mesmo que mat += 1, porém “mat” é um ponteiro fixo, definido na stack e não pode ser alterado.
c) p= ++mat; Inválido pelo mesmo motivo da questão b
d) x=(*mat)++; Válido, pois x é um inteiro qualquer e (*mat) é o valor armazenado em mat, ou seja, não está ocorrendo incremento no vetor em sí, mas sim no valor armazenado, dessa forma temos um incremento em um inteiro que está sendo retornado para outro inteiro.

7 - Código upado no git e enviado em zip.

8 - Código upado no git e enviado em zip.

9 - opção E) Minha explicação: A função while (*s1) s1++; percorre todo o vetor s1 até encontrar o fim do vetor, então, ao fim da execução do while, o s1 estará sendo apontado para o fim do vetor passado. Agora analisando o for, percebe-se que o vetor s2 é percorrido até o final, parando apenas ao encontrar \0, porém eles não apenas percorrem, já que o argumento ((*s1 = *s2)!=’\0’) está atribuindo os valores de s2 para s1 e ao mesmo tempo comparando com o \0.

10 - a) A função eleva n ao quadrado. A cada passada do laço for, j tem o valor somado com o valor anterior da variável i, que possui sempre um valor ímpar, basicamente o que ocorre é que a cada repetição do laço, j assume valores de números quadrados sequencialmente, ou seja, assume valores como 1, 4, 9, 16, 25, etc. E uma vez que j repete a quantidade de vezes necessária, o laço termina, mantendo o quadrado correto de n.

b) O resultado para um n = 10 é 100.
c) Código upado no git e enviado em zip.